İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

A COMPUTATIONAL MODEL FOR ANAPHORA RESOLUTION IN TURKISH BASED ON THE CENTERING THEORY

M.Sc. Thesis by

Ramiz Erman AYKAÇ

Department : Defence Technologies

Programme: Information Technologies

MAY 2005

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

A COMPUTATIONAL MODEL FOR ANAPHORA RESOLUTION IN TURKISH BASED ON THE CENTERING THEORY

M.Sc. Thesis by

Ramiz Erman AYKAÇ

514021102

Date of Submission : 9 May 2005
Date of Defence Examination : 02 June 2005

Supervisor (Chairman): Prof. Dr. Eşref ADALI

Members of the Examining Committee Assoc. Prof. SABİH ATALAN

Assoc. Prof. Elif KARSLIGİL

MAY 2005

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TÜRKÇE İÇİN MERKEZLEME TABANLI ANAFOR ÇÖZÜMLEMESİ YAPAN BİLGİSAYISAL BİR MODEL

YÜKSEK LİSANS TEZİ

Ramiz Erman AYKAÇ

514021102

Tezin Enstitüye Verildiği Tarih : 9 Mayıs 2005
Tezin Savunulduğu Tarih : 2 Haziran 2005

Tez Danışmanı : Prof. Dr. Eşref Adalı

Diğer Jüri Üyeleri Doç. Dr. Sabih ATALAN

Doç.Dr. Elif KARSLIGİL

MAYIS 2005

PREFACE

Natural Language applications, which are concerned and intensively researched by many researchers since many years, can make our lives easier in many different areas. The aim of these reaches, of course, is moving the present linguistic theories and findings to one step forward. As a computational aspect, it is tried to make these theories use in daily life easily and efficiently.

Generally so many studies for many natural languages in the world has been worked on. For Turkish, however, there are insufficient number of researches either as theoretical or computational. From my point of view; it is the main purpose of writing this thesis that, the computational studies on Turkish is very inadaquate, altghough it is very suitable to be modelled computationally.

Anaphora Resolution is very interesting and important subject in Natural Language Processing applications. For these reasons, in this study I have tried to develop an application on “Anaphora Resolution” process based on Turkish.

I would like to thank my supervisor Prof. Dr. Eşref ADALI for giving me all the necessary supports and valuable advices to complete this study. And also thanks to my dear friends Ress. Asst. Gülşen ERYİĞİT and Ress. Asst. A. Cüneyt TANTUĞ who guided me with their ideas and suggestions till summer. Finally, I want to extent special thanks to my dear friend Savaş YILDIRIM because of allocating his valuable time and giving his intensive efforts in means of helping for my study.

May 2005 R. Erman AYKAÇ

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

A1SG	: First Person Singular
A2SG	: Second Person Singular
A3SG	: Third Person Singular
A1PL	: First Person Plural
A2PL	: Second Person Plural
A3PL	: Third Person Plural
ACC	: Accusative case
DAT	: Dative case
GEN	: Genitive
ICP	: Initiating Conversation Participant
NI	: Noun Incorporation
NP	: Noun Phrase
NLP	: Natural Language Processing
OCP	: Another Conversation Participant
PLU	: Plural
POS	: Part Of Speech
PROG	: Progressive
SING	: Singular
SOV	: Subject – Object – Verb Order

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$C_f(U_n)$	: Forward looking centers of utterance “n”.
$C_b (U_n)$	: Backward looking centers of utterance “n”.
$C_p (U_n)$	: Preferred center of utterance “n”.
U_n	: Utterance “n”.
$P_a(a)$	: Possible antecedents of a pronoun “a”.

TÜRKÇE İÇİN MERKEZLEME TABANLI ANAFOR ÇÖZÜMLEMESİ YAPAN BİLGİSAYISAL BİR MODEL

ÖZET

Bu çalışmada Türkçe cümlelerdeki bazı adılların işaret ettiği varlıkların tespit edilmesini sağlayan bir bilgisayısal model dizayn edilmiştir. Bu modelin geliştirilmesinden önce konuya ilişkin Türkçe Dili üzerine yapılan buluş ve çalışmalar üzerine yoğun bir araştırma yapılmıştır.

Yapılan araştırmalar neticesinde böyle bir uygulama için kullanılabilecek bir yaklaşım olarak Merkezleme Teorisine dayalı bir algoritma temel olarak seçilmiştir. Bu yaklaşımı temel alan bu uygulamanın üreteceği sonuçların gözlemlenmesi amacıyla modeli oluşturan modullerin bir çoğu da gerçeklenmiştir.

Bu teori genel olarak cümleler içerisindeki baskın varlıkların belirli kurallarla bir sonraki cümleye taşındığını iddia eder. Bu teoriyi temel alan yaklaşımda, cümle içerisindeki adıllar, önceki cümlelere taşınan bu baskın varlıkları işaret etmektedir. Merkezleme teorisi cümleler içerisinde varlık belirten ifadeler arasında bir üstünlük hiyerarşisine ihtiyaç duyduğundan, böyle bir hiyerarşinin kullanılan dil için belirlenmesi gerekir.

Bu nedenle bu yaklaşımı temel alan önceki çalışmalarda ortaya atılan hiyerarşilerin Türkçe Dili için uygunluğu çeşitli deneylerle test edilmiştir. Bunun yanı sıra bu yaklaşımı temel almayan ancak benzer konularda baskınlığı inceleyen başka çalışmaların da bu amaç için kullanılabilirliği de incelenmiştir. Elde edilen ilk sonuçlarda tam bir uygunluk sağlayan hiyerarşi bulunamadığından bu çalışma içerisinde üstünlüğü belirleyecek bir hiyerarşi önerilmiş ve bu hiyerarşi üzerine el yordamıyla hazırlanmış veriler üzerinde bazı testler uygulanmıştır. Elde edilen son veriler incelendiğinde önerilen hiyerarşinin tatmin edici sonuçlar ürettiği gözlemlenmiştir. Önerilen bu hiyerarşi tasarlanan modelin gerçeklenen modülleri içerisinde kullanılarak, Türkçe için adıl çözümlemesi yapan bir uygulama geliştirilmiştir.

Genel olarak, tasarlanan bu uygulama modüler yapıda olduğundan daha sonradan geliştirilmeye uygundur. Ayrıca temelinde çok detaylı bilgileri ele almaya uygun olarak tasarlanan veri tabanı yapısı sayesinde, uygulamanın daha sonra sadece Anaphora Resolution değil başka alanlarda yapılacak dilbilim çalışmalarında da kullanılmaya uygun halde olduğu düşünülebilir.

A COMPUTATIONAL MODEL FOR ANAPHORA RESOLUTION IN TURKISH BASED ON THE CENTERING THEORY

ABSTRACT

In this study, a computational model which provides defining the pronouns and their antecedents in previous sentences is designed. Before designing the model, an intensive research on especially Turkish studies, theories and linguistic findings has been completed. As a result of these reseaches, an algorithm based on the Centering Theory has been thought suitable to use in this application. For the purpose of observing the results, many of the modules form the application have been implemented.

In general, The Centering Theory claims that the salient entities are carried to the next sentences in a discourse. A new approach which bases on this theory, claims that pronouns in the sentences actually references to more salient antecedents which are located in previous sentences. Since the Centering Theory needs an hierarchy to define the salience factor between the entities, it should be presented for the natural language to be used.

For this reason, the suitability of the presented hierarchies for Turkish based on this approach has been tested by a number of experiments. On the other hand, some other approaches which are not based on this approach but tries to define another type of hierachies has also been tested whether they can be used in this aspect. Since the sufficient success could not be achieved from the results of these tests, a new hierarchy is suggested and tested on a number of manually prepared data.

Having the final results, it is observed that the hierachy is completed the sufficient evaluations. Using this hierarchy in the completed modules, an application has been developed for Anaphora Resolution on Turkish Language.

Since the general structure of this model is modular, it is able to be improved easily. And also, it can be thought that the application can be used on some applications that are different than the anaphora resolution since the basic database is constructed with very detail segments.

1. Introduction

1.1 Introduction and the Purpose Of The Study

Natural language processing is very popular and it has been investigated since 70's. Some of these researches are on theorical linguistic studies and some are on computational models. Since the "Natural Language" branch involves in making the computers understand the natural language, the applications developed under this branch is distributed to many sub-branches. And also, it is needed to make very serious researches and studies on linguistic science before developing such applications. Hence, it can easily be inferred that the natural language subjects are really comprehensive.

In natural language studies, it is possible to develop either some specific applications on a strict subject or more general critical sub-applications that many other applications may need them. Morphological analizer or Pos tagger applications may be given as examples to such type applications. And additionally, one of the other example may be the Anaphora Resolution application.

Anaphora resolution can be defined as finding the antecedents of pronouns whose antecedents or referents are evoked in previous sentences in a discourse. Although this type of applications may be seem meaningless, they can be a subroutine for some specific applications. Machine translation and text summarizing can be given as the example.

This study involves the automatic anaphora resolution process which is one of the crucial task in natural language applications. This resolution process presented in the study is invastigated for many factors and languages but the implementation, however, is done for Turkish language. And also, the algorithm which the application based on, is the Centering Algorithm.

2. Natural Language Processing

2.1 The Definition

Since this study is a natural language processing in general, the best important thing is defining the concept of NLP. In Bar-Hillel (1971), the definiton of natural language processing is given as:

> " Natural Language Processing (NLP) is the use of computers to process written and spoken language for some practical, useful, purpose: to translate languages, to get information from the web on text data banks so as to answer questions, to carry on conversations with machines, so as to get advice about, say, pensions and so on. "

From this definiton we understand that the study of natural language processing deals with human and the computer reaction at all. The main purpose of this area of specialization is making the use of natural language by the help of computers. It can easily be inferred from this purpose that such a goal needs to handle many other topics in computer science. It is explained in the following definition taken from the same reference above:

> " NLP is not simply applications but the core technical methods and theories that the major tasks above divide up into, such as Machine Learning techniques, which is automating the construction and adaptation of machine dictionaries, modeling human agents' beliefs and desires etc. This last is closer to Artificial Intelligence, and is an essential component of NLP if computers are to engage in realistic conversations: they must, like us, have an internal model of the humans they converse with. "

Geman and Johnson (2000) defines that, natural language processing is the use of computers for processing natural language text or speech. Natural language interfaces permit computers to interact with humans using natural language, e.g., to query databases.

As observed in Roland (2001); the practical development of computers began around 1940. From then on, there evolved a basic distinciton between numerical and nonnumerical computer science.

Numerical computer science specializes in the calculation of numbers. In the fields of physics, chemistry, economics, sociology, etc., it has led to a tremendous expansion of scientific knowledge. Also many applications like banking, air travel, stock inventory, manufacturing, etc., depend heavily on numerical computation.

Without computers and their software, operations could not be maintained in these areas.

Nonnumerical computer science deals with the phenomena of perception and cognition. Despite hopeful beginnings, nonnumerical computer science soon lagged behind the numerical branch. In recent years, however, nonnumerical computer science has made a comeback as artificial intelligence and cognitive science. These new interdisciplinary fields investigate and electronically model natural information processing.

Roland (2001) claims that, the term “Natural Language Processing” which is very close with “Computational Linguistics” in definition, refers to that subarea of nonnumerical computer science which deals with language production and language understanding. Like artificial intelligence and cognitive science in general, computational linguistics is a highly interdisciplinary field which comprises large sections of traditional and theoretical linguistics, lexicography, psychology of language, analytic philosophy and logic, text processing, and the interaction with databases, as well as the processing of spoken and written language.

2.2 Language Science and Its Components

A speaker of Turkish knows meaning of a word like “kırmızı (red)”. When asked to pick the red object among a set of non-red objects, for example, a competent speaker-hearer will be able to do it. A standart computer, on the other hand, does not “understand” what red means, just as a piece of paper does not understand what is written on it.

According to Roland (2001), in the interaction with a standart computer, the understanding of natural language is restricted largely to the user. For example, if a user searches in a database for red object, he understands the word red before it is put into and after it is given out by the standart computer. But inside the standart computer, the word red is manipulated as a sign which is uninterpreted with respect to the color denoted.

What is true for standart computers does not apply to human-computer communication in general, however. Consider for example a modern robot which is asked by its master to get an object it has not previously encountered, for example, the new blue and yellow book on the desk in the other room. If such a robot is able to spontaneously perform an open range of different jobs like this, it has an understanding of language which at some level may be regarded as functionally equivalent to the corresponding cognitive procedures in humans.

The communication with a robot may be based on either artificial or natural language. The use of natural language is much more challenging, however, and much preferable in many situations. As a first step towards achieving unrestricted human-computer communication in natural language, it should be considered the current state of linguistics. In this field of research, three basic approaches to grammatical analysis may be distinguished, namely (i) traditional grammar, (ii) theoritical linguistics and (iii) computational linguistics. They differ in their methods, goals and applications. This difference between these topics is defined in Roland (2001) as following:

> " Traditional Grammar, uses the method of informal classification and description based on tradition and experience, has the goal to collect and classify the regularities and irregularities of the natural language is question as completely as possible and is applied mostly in teaching languages.
>
> Theoretical Linguistics uses the method of mathematical logic to describe natural languages by means of formal rule systems intended to derive all and only the well-formed expressions of a language which has the advantages of stating grammatical hypotheses explicitly, pursued the goal of describing the "innate human language ability", whereby aspects of language use in communication have been excluded, and has had rather limited applications because of its fragmentation into different schools.
>
> Computational Linguistics combines the method of traditional grammar and theoretical linguistics with the method of effectively verifying explicit hypotheses by implementing formal grammars as efficient computer programs and testing them automatically on realistic (very large) amount of data. It has the goal of modeling the mechanism of natural language communication which requires a complete morphological, lexical, syntactic, semantic and pragmatic analysis of a given natural language within a functional framework, and has applications in all instances of human-computer communication far beyond letter-based language processing. "

From the definitions above, we can say that traditional grammars is used in real life and created by the experience of humans. So as a result it is not followed pre-defined rules that are prepared or discussed before its creation. The theoretical linguistics, however, uses some rules defined in mathematics and models the facts found in the natural language. This modeling is used to cerate some computational applications of natural language. By creating these applications, the concept of computational linguistics arises finally.

As claimed in Ralph (1986), despite their different methods, goals, and applications, the three variants of language science described above divide the field into the same components of natural language processing, namely phonology, morphology, lexicon, syntax, semantics, and the additional field of pragmatics. The role played by

these components and the ways in which they are handled scientifically differs, however, within the three different approaches.

3. The Components of Natural Language Processing

In this section of the study, the main parts and concepts of natural language processing is investigated in briefly. This investigation has been done by giving the definitions and the purposes of these main parts. And also, giving the definitions, different aspects are taken from a sort of different sources.

3.1 Phonology

In general, Phonology is the science of language sounds. As explained in Bird (1994), Phonology is the study of the system of sounds that are relevant to meaning. As such, phonology stands at the interface between grammar and speech.
Ralph (1986) claims that, in computational linguistics, the role of phonology is marginal at best. One might conceive of using it in automatic speech recognition and synthesis, but the appropriate science is infact phonetics.

3.2 Morphology

We know that any natural language consist of many words. Many applications in the branch of NLP deals with these words which are the smallest parts of any language. So as a definition; the part of linguistic dealing with this phenomena is Morphology. In the field of morphology, the words of a language are classified according to their part of speech (category), and the structure of word forms is described in terms of inflection, derivation, and composition.
To traditional grammar, morphology has long been central, as shown by the many paradigm tables in, for example, grammars of Latin. In theoretical linguistics, morphology has played a minor role. Squeezed between phonology and syntax, morphology has been used mostly to exemplify principles of either or both of its neighboring components. In computational linguistics, morphology appears in the context of automatic word form recognition. It is based on an on-line lexicon and a morphological parser which relates each word form to its base form and characterizes its morpho-syntactic properties. Automatic word form recognition is presupposed by all other rule-based techniques of automatic language analyis, such as syntactic and semantic parsing (Roland ,2001).

3.2.1 Words

A word is defined as the set of word forms in its inflectional paradigm, as explained in Hillel (1971). According to this definition, a word is an abstract concept which is concretely manifested solely in the associated word forms. For the name of a word, its base form is used. The traditional base from of nouns is the nominative singular, e.g., kitap (book); of verbs it is the infinitive of the present tense, e.g., öğren (learn); and of adjectives it is the adnominal is the positive, e.g., yavaş (slow).
The words of a natural language are traditionally divided into the following parts of speech:

a) Verbs: walk, read, give ..

b) Nouns: book, table, woman, arena ..

c) Adjectives: quick, beautiful, good ..

d) Conjunctions: and, or- because, after ..

e) Prepositions: in, on, over, under, before ..

f) Determiners: a, the, every, some, all ..

g) Particles: only, already, just

The words of natural language are concretely realized as word forms. For example, the English word “write” is realized as the word forms write, writes, wrote, written, and writing. The grammatical well-formedness of natural language sentences depends on the choice of the word forms.

It is said in Fromkin and Rodman (1983) that, in traditional morphology, the following principles of combination are distinguished:

a) Inflection is the systematic variation of a word which allows it to perform different syntactic and semantic functions, and adapt to different syntactic environments. (e.g., learn, learns/s, learn/ed, learn/ing)

b) Derivation is the combination of a word with an affix. (e.g., clear/ness, clear/ly, un/clear)

c) Compounding is the combination of two or more words into a new word form (e.g., gas/light, hard/wood, over/indulge and over-the-counter)

The grammarians of ancient Greece and Rome arranged inflectional word forms into paradigms.

3.2.2 Turkish Morphology in General

As pointed out in Oflazer (2003), Turkish has agglutinative morphology with productive inflectional and derivational suffixations. The number of word forms one can derive from a Turkish root form may be in the millions (Hankamer ,1989). The number of possible word forms that can be obtained from a NOUN, a VERB, and an ADJECTIVE root form by suffixing 1, 2 and three morphemes is listed in Table 3.1. As shown in Oflazer (2003), it is possible to obtain 33 different surface words by suffixing only one morpheme to a noun.

Table 3.1: The number of possible word formations obtained by suffixing 1,2 and 3 morphemes to a Noun, Verb and an Adjective

CATEGORY	**Number Of Morphemes**		
	1	2	3
NOUN	33	490	4.825
VERB	46	895	11.313
ADJECTIVE	32	478	4.789

The number of words in Turkish is theoretically infinite, since there is no syntactic limit on the number of derivational suffixes a word can take. For example, it is possible to embed multiple causatives is a single word (as in: somebody causes some other person to cause another person … to do something). For example:

(3.1)

Yapmak, Yaptırmak, Yaptırttırmak, Yaptırtırtırmak, …

And it is also discussed according to the word order form of Turkish in Oflazer (2003). Turkish is a free constituent order language, in which constituents at certain phrase levels can change order rather freely according to the discourse context or text flow. The typical order of the constituents is subject-object-verb (SOV), however, other orders are also common, especially in discourse.

The morphology of Turkish enables morphological markings on most of the constituents to signal their grammatical roles without relying on their order. This does not mean that the word order is not important, sentences with different word orders reflect different pragmatic conditions, that is the topic, focus, and backgound information conveyed by those sentences differ. Word order inside the noun phrases is more constrained, with specifiers preceding modifiers, but within each group, order (e.g., between cardinal and attributive modifiers) is mainly determined by which aspect is to be emphasized. For instance “iki genç adam” (two young men) and “genç iki adam” (young two men) are both possible in Turkish, but the former

being the neutral case or the case where youth is emphasized, while the latter is the case where the cardinality is emphasized.

3.3 Lexicon

Lexicon can be defined as listing analyzed words. The words of a language are collected and classified in lexicography and lexicology. Lexycography deals with the principles of coding and structuring lexical entries, and is a practically oriented border area of natural language science. Lexycology investigates semantic relations in the vocabulary of a language and is part of traditional philology.

In computational linguistics, electronic lexical combine with morphological parsers in the task of automatic word form recognition. The goal is maximal completeness with fast access and low space requirements. In addition to building new lexical for the purpose of automatic word form recognition, there is great interest utilizing the knowledge of traditional lexical for automatic language processing (mining of dictionaries).

3.4 Syntax

In communication, according to Roland (2001), the task syntax is the composition of meanings via the composition of word forms. One aspect of this is characterizing well-formed compositions in terms of grammatical rules. The other is to provide the basis for a simultaneous semantic interpretation.

Hillel (1971) explains that, syntax analysis performs two main functions in analyzing natural language input:

Determining the the structure of the input. In particular, syntax analysis should identify the subject and objects of each verb and determine what each modifying word or phrase modifies. This is most often by assigning a tree structure to the input, in a process referred to as parsing.

Regularizing the syntactic structure. Subsequent processing (i.e., semantic analysis) can be simplified if we map the large number of possible input structures into a smaller number of structures.

In theoretical linguistics, syntactic analysis has concentrated on a description of grammatical well-formedness. The problem with analyzing well-formedness in isolation is that any finite set of sentences may be described by a vast multitude of different grammars. In order to select the one type of description which turns out to

be correct in the long run, theoretical linguistics has vainly searched for universals supposed to characterize the “innate human language faculty”.

A more realistic and effective standart is the requirement that the grammar must be suitable to serve as a component in an artificial cognitive agent communicating in natural language. Thereby, the descriptive and functional adequacy of the grammar may be tested automatically on the full range of natural language data. This presupposes a grammatical algorithm with low mathematical complexity. Furthermore, the algorithm must be input-output equivalent with the mechanism of natural language communication (Roland ,2001).

3.5 Semantics

Semantics, in other words science of literal meanings of natural language may be divided into lexical semantics, describing the meaning of words, and compositional semantics, describing the composition of meanings in accordance with the syntax. The task of semantics is a systematic conversion of the syntactically analyzed expression into a semantic representation based on the functor-argument structure underlying the categories of basic and complex expressions.

The beginning of traditional grammar contributed considerably to the theory of semantics, for example Aristotle’s distinction between subject and predicate. However, these contributions were passed on and developed mostly within philosophy of language. In traditional grammar instruction, the treatment of semantics did not reach beyond the anecdotal.

Semantics was initially limited to characterizing syntactic ambiguity and paraphrase, in theoretical linguistics. Subsequently, logical semantics was applied to natural language based on a metalanguage, natural language meanings were defined in terms of truth conditions.

Computational linguistics uses procedural semantics instead of metalanguage-based logical semantics. The semantic primitives of procedural semantics are based on operations of perception and action by the cognitive agent. The semantics is designed to be used by the pragmatics inan explicit modeling of the information transfer between speaker and hearer.

3.6 Pragmatics

Pragmatics is the study of how grammatically analyzed expressions are used relative to the context of interpretation. Therefore, pragmatics is not part of the

grammar proper, but concerned with the interaction between the expressions and the context, presupposing the grammatical analysis of the expressions and a suitable description of the context.

In traditional grammar, phenomena of pragmatics have been handled in the separate discipline. This has been obstacle to integrating the analysis of language structure and language use.

In theoretical linguistics, the distinction between semantics and pragmatics has evolved only haltingly. Because theoretical linguistics has not been based on a functional model of communication, pragmatics has served mostly as the proverbial “wasebasket” (Hillel,1971).

In computational linguistics, the need for a systematic theory of pragmatics became most obvious in natural language generation as in dialogue systems or machine translation, where the system has to decide what to say and how to say it in a theorically acceptable way.

That the different approaches of traditional grammar, theretical linguistics, and computational linguistics use the same set of components to describe the phenomena of natural language, despite their different methods and goals, is due to the fact that the division of phenomena underlying these components is based on different structural aspects, namely sounds (phonology), word forms (morphology), sentences (syntax), literal meanings (semantics), and their use in communication (pragmatics).

4. Computational Linguistics

In Ralph (1986), it is defined that the computational linguistic is the study of computer systems for understanding and generating natural language. The definition and the main parts of the Natural Language Science is discussed above. And also this discussion has been made according to different aspects of linguistics. Since this study serves a computational model of linguistic subject as Anaphora Resolution, detailed observation of Computational Linguistic is also necessary.

As given in Geman and Johnson (2000), computational linguistics studies the computational processes involved in language learning, production and comprehension. Computation linguists believe that the essence of these processes (in humans and machines) is a computational manipulation of information.

It can easily be inferred that natural language is an integral part of our lives. Language serves as the primary vehicle by which people communicate and record information. It has the potential for expressing an enormous range of ideas, and for conveying complex thoughts succinctly. Because it is so integral to our lives, however, we usually take its powers and influence for granted.

The aim for computational linguistics is, in a sense, to capture this power. By understanding language processes in procedural terms, we can give computer systems the ability to generate and interpret natural language. This would make it possible for computers to perform linguistic tasks (such as translation), process textual data (books, journals, newspapers), and make it much easier for people to access computer-stored data. A well-developed ability to handle language would have a profound impact on how computers are used. (Ralph (1986))

4.1 Methods and Applications of Computational Linguistics

Computational linguistics uses parsers for the automatic analysis of language. According to Roland (2001), the term parser is derived from the Latin word pars meaning part, as is part of speech. Parsing in its most basic form consist in the automatic decomposition of a complex sign into its elementary components, classification of the components via lexical lookup, and composition of the classified

components via syntactic rules in order to arrive at an overall grammatical analysis of the complex sign.

Methodologically, the implementation of natural language grammars as parsers is important because it allows one to test the descriptive adequacy of formal rule systems automatically and objectively on real data. This new method of verification is as characteristic for computational linguistics as the method of repeatable experiments is for natural science. Practically, the parsing of natural language may be used in different applications.

Practical tasks of computational libguistics may be classified as below as given in Roland (2001):

> "Indexing and retrieval in textual databases
>
> Textual databases electronically store texts such as publications of daily newspapers, medical journals, and court decisions. The user of such a database should be able to find exactly those documents and passages with comfort and speed which are relevant for the specific task in question. The World Wide Web (www) may also be viewed as a large, unstructured textual database, which daily demonstrates to a growing number of users the difficulties of succesfully finding the information desired.
>
> Machine Translation
>
> Especially in the European Union, currently with eleven different languages, the potential utility of automatic or even semi-automatic translation systems is tremendous.
>
> Automatic text production
>
> Large companies wihic contunally birng out new products such as engines, video recorders, farming equipment, etc., must constantly modify the assciated product descriptions and maintenance manuals. A similar situation holds for lawyers, tax accountants, personnel officers, etc., who must deal with large amounts of corrependence in which most of the letters differs only in a few, well-defined places. Here techniques of automatic text production can help, ranging from simple templates to highly flexible and interactive systems using sophisticated linguistic knowledge.
>
> Automatic text checking
>
> Applications in this area ranges from simple spelling checkers (based on word form lists) via word form recognition (based on an morphological parser) to syntax checkers based on syntactic parsers which can find errors in word order, agreement, etc.
>
> Automatic Content Analysis
>
> The printed information on this planet is said to double every 10 years. Even in specialized field such as natural science, law or economics, the constant stream of relevant new literature is so large that researchers and professionals do not nearly have enough time to read it all. A reliable automatic content analysis in the form of brief summaries would be very useful. Automatic content analysis is also precondition for concept-based indexing, needed for accurate retrieval from texture databases, as well as adequate machine translation.

Automatic tutoring

There are numerous areas of teaching in which much time is spent on drill exercises such as the more or less mechanical practicing of regular and irregular paradigm in foreign languages. These may be done just as well on the computer, providing the students with more fun (if they are presented as a game for example) and the teacher with additional time for other, more sophisticated activities such as conversation. Furthermore, these systems may produce automatic protocols detailing the most frequent errors and the amount of time needed for various phases of exercise. This contitutes a valuable heuristics for improving the automatic tutoring system ergonometrically. It has led to a new field of research in which the electronic text book of old is replaced by new teaching programs utilizing the special possibilities of the electronic medium to facilitate learning in ways never explored before.

Automatic dialog and information systems

These applications range from automatic information services for train schedules via queries and storage in medical database to automatic tax consulting. This list is by no means complete, however, because the possible applications of computational linguistics include all areas in which humans communicate with computers and other machines of this level, today or in the future.”

5. Anaphora Resolution

5.1 Basic Terminology

Turanlı (1996) claims that the term utterance is considered to be an expression uttered or written by a particular speaker or writer at a particular time and at a particular location. Utterances thus contrast with sentences which are possible abstract constructs not situated in time and space.

A discourse may be thought as the sort of utterances in general. It consist of two or more sets of utterances that are coherently linked and also it may be written or spoken language. Discourse involves an initiating [conversation] participant (ICP) and another [conversation] participant (OCP) (Grosz and Sidner 1986). An ICP is the speaker or the writer who starts the discourse. The OCP, on the other hand, is the addressee, i.e. the hearer or the reader.

According to Jurafsky (2000), language does not normally consist of isolated, unrelated sentences, but instead of collocated, related groups of sentences. So this group of sentences is referred as discourse.

Manning and Schütze (1990) claims that, studies of discourse seek to elucidate the covert relationships between sentences in a text. In a narrative discourse, one can seek to describe whether a following sentence is an example, an elaboration, a restatement, etc. In a conversation one wants to model the relationship between turns and the kinds of speech acts involved (questions, statements, requests, acknowledgements, etc.).

As we discussed in chapter 3, in communication of “normal circumstances” in written or spoken language form we do not use the sentences phrases or words unrelated with each other. In other words there are coherence in such communication actions in general. Cohesion is the accounting phenomenon of this coherence. Cohesion occurs where the interpretation of some element in the discourse is dependent on that of another and involves the use of abbreviated or alternative linguistic forms which can be recognized and understood by the hearer or the reader, and which refer to or replace previously mentioned items in the spoken or written text. Consider the following example:

(5.1)

Ali_i yolda $Ahmet_j$ ile karşılaştı.

Ali road.LOC Ahmet with meet.REF.PAST.3SG

(Ali met with Ahmet on the road)

(5.2)

[Oi] Ona_j dün nerede kaldığını sordu.

He.ACC.3SG yesterday where stay.PAST.3SG ask.PAST.3SG

(He asked him where he stayed yesterday)

It is normal to assume that the second sentence is related to the first one and that he refers to Ali. This reference show that there is a cohesion between these two sentences. If we change (5.2) with “Bu bilgisayar çalışmıyor.” (This computer is not working.) there will be no cohesion. Because there will be no relation between (4.1) and (5.2) in this case.

Jurafsky and Martin (2000) claims that a natural expression used to perform reference is called a “referring expression”, and the entitty that is referred to is called the “referent”. Two referring expressions that are used to refer to the same entity are said to corefer. By using this definition, consider the following example:

(5.3)

Ali_i galeride çok güzel bir $araba_k$ gördü. [O_i] Onu_k satın almak istedi.

Ali gallery.LOC very beatiful car see.PAST.3SG. It.ACC buy want.PAST.3SG

Ali saw a very beatiful car in the gallery. [He] wanted to buy it.

In (5.3) above, “araba” and “Onu” are referring expressions and araba is their referent. And also, araba and onu are corefer in this example.

Halliday and Hasan (1976) describe anaphora as ‘cohesion which points back to some previous item’. The ‘pointing back’ word or phrase is called anaphora and the entity to which it refers or for which it stands is its antecedent. The process of determining the antecedent of an anaphora is called “anaphora resolution”. When the anaphor refers to an antecedent and when both have the same referent in the real world, they are termed “coreferential”. It can easily be understood the meaning

of these definitions by assuming the example above. In (5.2) “Ona” is an anaphora and Ahmet is its antecedent. And Ona and Ahmet is coreferential.

Examples of coreference, which is the act of picking out the same referent in the real world is introduced above. Sometimes, as seen in the example below, a specific anaphor and more than one of the preceding (or following) noun phrases may be coreferential thus forming a coreferential chain of entities which have the same referent

(5.4)

Istanbul’da$_i$ yaşamayı çok seviyorum. Burada$_i$ herşey çok farklı. Bir gün buradan$_i$ ayrılacağımı hiç sanmıyorum.

Istanbul.LOC live.ACC very like.PROG.1SG. Here.LOC everything very different. One day here.ABL live.FUT.ACC.1SG. never think.NEG.PROG.1SG

I like living in Istanbul too much. Here, everything is very different. I don’t think that I will leave here one day.

Coreferential chains partition discourse entities into equivalence classes. On the other hand, there may be cases where two items are coreferential without being anaphoric. Cross-document coreference is an obvious example such that; two mentions of the same person in two different documents will be coreferential, but will not stand in anaphoric relation.

As stated in Özgür (1997),

> “Anaphors, whose denotations are traditionally assumed to be referring expressions, are typically employed for discourse-salient reference to related entities denoted by noun phrases.”

From this definiton, we understand that the anaphors are actually related with some entites that are more salient than the others. The function of this salience will be discussed in the following sections in this study. But this relation between the salient antecedent and the anaphor is really important especially for the anaphora resolution process. This case will be discussed in section 7 of this study.

Cataphora is another concept in anaphora resolution. It arises when a reference i made to an entity mentioned subsequently in the text.

(5.5)

O$_i$ şimdi ünlü bir şarkıcı.

She is now famous one singer

(She is now a famous singer.)

(5.6)

Ama Ayşe'nin$_i$ böyle bir şarkıcı olacağını zaten bekliyorduk.

But Ayşe.GEN such singer be.FUT.AC already wait.PAST.PROG.3PLU

(But [we] were already waiting that she would be such a singer.)

In this example "O" refers to Ayşe, mentioned subsequently. Cataphora is similar to anaphora, the difference being the direction of the pointing.

Manning and Schütze (1990) states that anaphora resolution is the central problem in a discourse. According to that reference, one way to approach lexical semantics is to study how word meanings are related to each other. Anaphoric relations hold between noun phrases that refer to the same entity in a sentence.

According to Jurafsky and Martin (2000), the set of referential phenomena that natural language provide is quite rich. The types of five basic referring expressions are surveyed in that study, which are indefinite noun phrases, definite noun phrases, pronouns, demonstratives, and one-anaphora. Actually all of these expressions are not exist in Turkish. Since this study is an implementation for Turkish language, we do not need to invastigate these expressions for English in detail. So, a detailed discussion of the reference phenomena for Turkish is given in chapter 5.1 of this study.

We should also illustrate another concept that is related with the discourse and anaphors. When the antecedent is a noun phrase (NP), it becomes convenient to abstract away from its syntactic realization in order to capture certain subtitles of its semantics. The abstraction termed a "discourse entity", allows the NP to be modeled as a set of one or more elements and provides a natural metaphor for describing what may on the surface seem to be grammatical number conflicts. Consider the following exapmle:

(5.7)

Ayşe siyah gökyüzündeki yıldızların o gece nasıl göründüklerini ancak görebiliyordu.

Ayşe black sky.DAT star-PLURAL.GEN that night how look.PAST. PASSIVE.ACC almost see.PAST.

(Ayşe could almost see the stars in the black sky, how they had looked that night.)

The discourse entity described by the noun phrase “Ayşe” consists of one element – the specific person in question, whereas the discourse entity represented by the noun phrase the stars incorporates all the stars in the sky that Ayşe could ‘almost see’. Now consider the following example:

(5.8)

Öğretmen her çocuğa pastel verdi. Onlar da rengarenk resimler çizmeye başladı.

The teacher each child-DAT crayon give-PAST. They also colorful picture-PLURAL draw start-PAST

(The teacher gave each child a crayon. They started drawing colourful pictures.)

The discourse entity represented by the noun phrase “her öğrenci (each child)” comprises all children in the teacher’s class and is therefore referred to by a plural anaphor. There also some cases that the antecedent of the plural anaphor is singular:

(5.9)

Ahmet’in ailesi her yaz Almanya’yı ziyaret eder. Bu yaz da gidecekler.

Ahmet-GEN family every summer Germany-ACC visit. (They*) this summer also go-FUT-PLU.

(Ahmet’s family visit Germany every summer. They will go also this summer.)

If the discourse entity associated with the NP “aile (the family)” is now considered, it is easy to explain the number ‘mismatch’: this discourse entity as a set contains more than one element. Therefore, the anaphora agrees with the number of the discourse entity associated with its antecedent rather than the number of the NP representing it. For the sake of simplicity, it is limited the treatment of the antecedent to its classical definition as a linguistic form (e.g. surface constituent such as noun phrase) and, therefore, retrain from searching for an associated discourse entity (e.g. semantic set). This is an approach widely adopted by a number of anaphora resolution systems that do not have recourse to sophisticated semantic analysis. It should be borne in mind, however, that there are cases where more detailed semantic description or processing is required for the successful resolution.

5.2 Requirements for Computational Anaphora Resolution

In this chapter the sources of knowledge needed for anaphora resolution is discussed. It is introduced the different phases of the pre-processing and resolution process and explained what tools and resources are necessary.

The disambiguation of anaphors is a challenging task and considerable knowledge is required to support it from low level morphological and lexical information to high level semantic and pragmatic rules.

5.2.1 Morphological and Lexical Knowledge

Morphological and lexical knowledge is required not only for identifying anaphoric pronouns, but also as input to further syntactic processing. In English, some anaphors are successfully resolved solely on the basis of lexical information such as gender and number. The fact that the nominal anaphors usually match their antecedents in gender and number sometimes sufficient for singling out a uniquie NP candidate as in the following example:

(5.10)

Ahmet had no letters from Ayşe while in İstanbul. He feared silence.

As we see in this example, he may refer to Ahmet or Ayşe. But since the reference of he must be a male, by using the gender matching rule Ahmet will be choosen. Because the remaining candidates “Istanbul”, “Ayşe” and letters are discounted on the basis of gender or number mismatch. The case is not so easy in Turkish since the pronouns have not got a gender slot in Turkish language. But anyway, number slot is also available for Turkish pronouns. Consider the example:

(5.11)

Ahmet$_i$ arkadaşlarını$_k$ sever.

Ahmet friend.POSS.PLU.ACC very like

(Ahmet likes his friends.)

(5.12)

[O$_i$] Onları$_k$ her zaman arar.

They.ACC every time call.AOR

(He call them always.)

In the example above, the pronoun onları (they.ACC) is plural. So the number matching will discount the singular antecedents, which is Ahmet in this example.

Gender agreement is useful criterion in English when the candidates for the anaphor are:

a) proper female or male names such as Jade, John, etc.,

b) nouns referring to humans such as man, woman, father, mother, son, daughter, etc.,

c) nouns representing professions such as teacher, doctor, singer, actor, actress which cannot be referred to by “it”,

d) gendered animals such as “cow“or “bull”,

e) words such as “country” or “ship” which can be referred to by either “she” or “it”.

In Turkish, however, gender is not a useful criteria since there is no such a property in pronouns of this language.

Number agreement helps to filter out candidates that do not carry the same number as the anaphor. It is the number of the discourse entity associated with each candidate (and anaphor in the case of definite descriptions) which is taken into account and not the number of the NP head. Coordinated antecedents “Erman ve Çağlar” are referred to by plural pronouns whereas collective nouns such as aile (family), takım (team), etc., can be referred to by both they and it.

In some languages the plural pronouns mark the gender (e.g. ils, elles in French, ellos, ellas in Spanish) and when a coordinated antecedent features both masculine and feminine. The above examples so far show that it is vital for an anaphora resolution system to have information not only about the gender and number of common nouns, but also about the gender and number of proper names.

5.2.2 Syntactic Knowledge

In the previous section of the study, the importance of morphological and lexical knowledge for the resolution process is demonstrated. In addition and more significiantly, the examples above show the importance of syntactic knowledge. Thus, in (5.9), “Ahmet” and “arkadaşlarını” should be identified as noun phrases. Therefore, it becomes clear that syntactic information about the constituents of the sentences is essential.

Syntax is indispensable in anaphora resolution. In addition to providing information about the boundaries of the sentences, clauses and other constituents (e.g. NPs, PPs), syntax plays an important role in the formulation of the different rules used in the resolution process. As an illustration, consider the simplified rule stipulating that an anaphoric NP is only coreferential with the subject NP of the same simple sentence or clause when the anaphor is reflexive as shown in (5.13) below:

(5.13)

Ahmet$_i$ bu işi kendisi$_i$ için yapıyor.

Ahmet this job.ACC himself for do.PROG.

(Ahmet is running this job for himself.)

This rule which relies on syntactic information about sentence or clause boundaries, along with information of about the syntactic function of each word, would rule out Ahmet as antecedent of “onun” (him) in (5.14):

(5.14)

Ahmet bu işi onun için yapıyor.

Ahmet this job.ACC he.GEN.3SG for do.PROG.

(Ahmet is running this job for him.)

Syntactic knowledge is used extensively in anaphora resolution and together with morphological and lexical knowledge it plays a key role in the process of anaphora resolution.

5.2.3 Semantic Knowledge

However important morphological, lexical and syntactic knowledge are, there are many cases where they alone cannot help to resolve anaphors. Consider the following example:

(5.15)

Ağaçta donakalmış olan kedi aşağı inmek istemedi.

Tree.LOC petrify.PAST.3SG be kitten below.ACC come down want.NEG.PAST.PROG

(The petrified kitten refused to come down from the tree.)

(5.16)

Aşağıda kendisine bakanlara şaşkınca baktı.

Below.LOC it.POSS.ACC look.PLU.ACC confusedly look.PAST

(It gazed confusedly at the onlookers below.)

In this example gender or number agreement rules can eliminate neither “donakalmış kedi” (petrified kitten) nor “ağaç” (the tree) as a potential antecedent because both candidates are gender in neutral. The selectional restrictions of the verb “bakmak” (to gaze) require that its agent (the subject in an active voice sentence) be animate; semantic information on the animacy of kitten would be crucial. In a computational system such information would reside in a knowledge base such as dictionary or ontology.

In some cases the correct interpretation of anaphors may depend on the ability of a system to undertake semantic processing in order to identify the discourse entity that is associated with the antecedent. Consider the following examples:

(5.17)

Her çocuk bisküvi yedi. Onlar çok lezzetliydi.

Every child biscuit eat.PAST.3SG. They very delicious.PAST

(Each child ate a biscuit. They were delicious.)

(5.18)

Her çocuk bisküvi yedi. Onlar çok şirinlerdi.

Every child biscuit eat.PAST.3SG. They very delight.PAST.3PLU.

(Each child ate a buscuit. They were delighted.)

In (5.17) above, the anaphor agrees with the number of the discourse entity associated with the antecedent bisküvi (biscuit). This plural discourse entity can be deduced from the quantifier structure of the sentence containing the antecedent. To this end, translation into logical form is necessary.

The examples above strongly suggest that a strategy of activating an anaphora resolution algorithm after semantic analysis rather than after syntactic analysis (parsing) will produce more accurate results. As a result, semantic knowledge is of

particular importance when interpreting lexical noun phrase anaphora, especially the indirect type. Therefore, considerable world knowledge and inferencing might be needed to determine the degree of compatibility of the modifiers.

5.2.4 Discourse Knowledge

Although the morphological, lexical, syntactic and semantic criteria for antecedent selection are very strong, they are still not always sufficient to distinguish among a set of possible candidates. Moreover, they serve more as filters to eliminate unsuitable candidates than as proposers of the most likely candidate. In the case of antecedent ambiguity, it is the most salient among the candidates for antecedent that is usually front-runner. This most salient element is referred to in computational linguistics as the “focus” (Grosz 1977 and Sidner 1979) or “center” (Grosz 1983, Joshi and Weinstein 1981).

As an illustration, neither machines nor humans would be confident in interpreting the anaphoric pronoun it in the sentence:

(5.19)

Tilly tried on the dress over her skirt and ripped it.

However, if this sentence wre part of a discourse segment, which would make it possible to identify the most salient element, the situation would be different:

(5.20)

Tilly’s mother had agreed to make her a new dress for the party. She worked hard on the dress for weeks and finally it was ready for Tilly to try on. Impatient to see what it would look like, Tilly tried on the dress over her skirt and ripped it.

In this discourse segment, dress is the most salient entity and is the center of attention throughout the discourse segment.

It is now clear that very often when two or more candidates compete for the antecedent role, the task of resolving the anaphor can be shifted to the task of tracking down the center/focus of the sentence or clause .

5.3 Computational Anaphora Resolution: Basic Steps

The automatic resolution of anaphors consist of the following main stages:

a) Identification of anaphors

b) Location of the candidates for antecedents

c) Selection of the antecedent from the set of candidates on the basis of anaphora resolvent factors.

These stages given above are common for all methods in anaphora resolution. The methods choosen for the operation, however, may differ in the 3rd stage. We have to point out that, this study is mainly tries to find an efficient method for Turkish anaphora resolution process. The details and the experiments for such methods are given in section 7 and 8. In the following sections of the study, this three main stages will be discussed. İn detail.

5.3.1 Identification of Anaphors

The first step in the automatic anaphora resolution is the identification of the anaphors whose antecedents have to be tracked down. The automatic identification of anaphoric words or phrases, as far as Turkish is concerned, is not a trivial task.

In pronoun resolution only the anaphoric pronouns have to be processed further, therefore for languages like English, non-anaphoric occurences, especially for “it”, have to be recognized by the program. For example:

(5.21)

It must be stated that Ahmet had passed the exam.

(5.22)

It is raining.

Fortunately, such cases is not possible for Turkish language. Because there is no anaphor that may be in non-anaphoric position in Turkish.

Even though most uses of first and second person pronouns as, ben (I), sen (you.2SG), biz (we), siz (you.2PL) are not anaphoric, their anaphoric use in reported speech or dialogue is not uncommon. The following example illustrates anaphoric uses of ben (I) (referring to Tülay Hanım). Simple rules for the identification of anaphoric first and second person pronouns include recognizing the text as reported speech or dialogue, and gender and number matching applied to potential anaphors or antecedents.

(5.23)

Ahmet Bey_i eve geldiğinde çok yorgundu.

Mr. Ahmet home.DAT come.PAST.LOC very tired.3SG

(Mr. Ahmet was very tired when he came home.)

(5.24)

[O_i] Odaya girince, $bana_k$ döndü ve “Gunaydın Tulay $Hanım_k$” dedi.

Room.DAT enter.WHEN i.DAT turn.PAST.3SG and “Good morning Mrs. Tülay”

say.PAST.3SG.

(As he entered the room, he turned me and said “Good morning Mrs. Tülay”.)

The search for anaphoric noun phrases can be even more problematic. Definite noun phrases (definite descriptions) are potentially anaphoric, often referring back to preceding noun phrases. For example:

(5.25)

$Ahmet_i$ iyi bir basketbol oyuncusuydu.

Ahmet good one basketball player.PAST.3SG

(Ahmet was a good basketball player.)

(5.26)

$Adam_i$ oldukça uzundu.

The man very tall.PAST.3SG

(The man was very tall.)

It is important to bear in mind that not every definite noun phrase is necessarily anaphoric. The discussion about definiteness and referentiality for Turkish can be found in Erguvanlı (1984), Dede (1986). And a detailed observation and discussion of the findings from Turanlı (1996) is also presented in section 6 of this study.

Morphological or lexical information is usually provided by a morphological analyser, part-of-speech tagger or dictionary. The advantage of a POS tagger is that it can disambiguate words that can be assigned more than one lexical category. However,

there are number of languages for which there are no POS tagger available. Therefore, programs for anaphora resolution in such languages have no choise but to use enhanced morphological analyser.

The detection of NP anaphors requires at least partial parsing in the form of NP extraction. A named entity recognizer, and in particular a program for identifying proper names, could be of great help at this stage. Zero anaphor identification requires more complete parsing, which reconstructs elliptically omitted items.

5.3.2 Location of the Candidates for Antecedents

Once the anaphors have been detected, the program has to identify the possible candidates for their antecedents. The vast majority of systems only handle nominal anaphora since processing anaphors whose antecedents are verb phrases, clauses, sentences or sequences is a more complicated task. Typically in such systems all noun phrases preceding an anaphor within a certain search scope are initially regarded as candidates for antecedents.

The search scope takes a different form depending on the processing model adopted and may vary in size depending on the type of anaphor. Since anaphoric relations often operate within or are limited to a discourse segment, the search scope is often set to the discourse segment that contains the anaphor (Kennedy and Boguraev, 1996). Anaphora resolution systems which have no means of identifying the discourse segment boundaries usually set the search scope to the current and N preceeding sentences, with N depending on the type of the anaphor. As Mitkov (1988) claimed, for pronominal anaphors, the search scope is usually limited to the current and two or three preceeding sentences.

Once all noun phrases in the search scope have been identified, different anaphora resolution factors are employed to track down the correct antecedent.

A ful parser can be used for identifying both noun phrases and sentence boundaries. However, it is possible to make do with simpler tools such as sentence splitter to single out consecutive sentences, and a noun phrase extractor to retrieve potential candidates for antecedents. A tokeniser is responsible for detecting independent tokens in the text, such as words, digits and punctuation marks. Several knowledge-poor approaches use POS taggers.

5.3.3 The Resolution Algorithm

Once the anaphors have been detected, the program will attempt to resolve them by selecting their antecedents from the identified sets of candidates. The resolution rules based on the different sources of knowledge and used in the resolution

process (as part of the anaphora resolution algorithm), are usually referred to as anaphora resolution factors. Factors frequently used in the resolution process include gender (generally not for Turkish language) and number aggreement, semantic (selectional) restrictions, salience, etc. These factors can be eliminating, i.e. discarting certain noun phrases from the set of possible candidates, such as in the case of gender and number constraints and selectional restrictions. The factors can also be “preferential”, giving more preference to certain candidates over others, such as salience (center of attention).

As a result, choosing a suitable resolution algorithm based on these factors is also very important part of the resolution system. These choice should be done by assuming the language itself. In this study the main algorithm is based ob the Centering Theory which will be told in detail in the following sections. By making this decision the simplicity and nature of the Turkish language is taken into account.

5.4 Theories and Algorithms for Anaphora Resolution Used In This Study

In this chapter some of the theories and algorithms that have been successfully used in anaphora resolution are outlined. These methods and algorithms includes the applied methods of this study. Some other approaches is also discussed in section 5.5.

5.4.1 Centering Theory and Centering Algorithm

Centering is a theory about discourse coherence and is based on the idea that is utterance features a typically most prominent entity called center. Centering regard utterances which continue the topic of preceding utterances as more coherent than utterances which feature topic shift.

The main idea of centering theory given in Grosz et al. (1983) and Grosz et al. (1995) is that certain entites mentioned in an utterance are more central than others and this imposes certain constraints on the use of referring expressions and in particular on the use of pronouns.

Discourses consist of continous discourse segments. A discourse segment D consist of a sequence of utterances U_1, U_2, ... , U_N. Each utterance U and D is assigned a set of potential next centers known as “forward-looking centers” $C_f(U, D)$ which corresponds to the discourse entities evoked by the utterance. Each utterance but the first one, is assigned a single center defined in the centering theory as the backward-looking centers $C_b(U)$ is a member of the set $C_f(U, D)$ and is the discourse entity the utterance U is about. The C_b entity connects the current

utterance to the previous discourse. It focuses on an entity that has already been introduced. A central claim of centering is that each utterance has exactly one backward-looking center.

The set of forward-looking centers $C_f(U, D)$ is partially ordered accoridng to their discourse salience. As pointed out in Brennan (1987), the highest-ranked element in $C_f(U)$ is called the "preferred center" $C_p(U)$. The preferred center in a current utterance U_N is the most likely backward-looking center of the following uttearnce U_{N+1}. We can say that, this ranking process is very complicated and depends on the language used for the resolution. Turkish remains to be a poorly investigated domain with respect to such ordering. This discussion and choosen hierarchy can be found in section 8 of this study.

A self-contained centering algorithm is offered by Grosz et al. (1995). Below are the rules and the algorithm of the Centering Theory presented in this study:

> **"Rule 1**: If any element of $Cf(U_n)$ is realized by a pronoun in utterance U_{n+1}, then $C_b(U_{n+1})$ must be realized as a pronoun also.
>
> **Rule 2**: Transition states are ordered. Continue is preferred to Retain. Retain is preferred to Smooth Shifting. Smooth Shifting is preferred to Rough Shifting. "

Rule 1 stipulates that if there is only one pronoun in an utterance, then this pronoun should be the backward-looking center. It is reasonable to assume that if the next sentence also contains a single pronoun- then the two pronouns corefer.

Rule 2 provides an underlying principle for coherence of discourse. Frequent shift detract from local coherence, whereas continuation contributes to coherence. Maximally coherent segments are those which do not feature changes of center, concentrate on one main discourse entity (topic) only and therefore require less processing effort. Rule 2 is used as a preference in anaphora resolution.

The algorithm given in in Grosz et al. (1995) is:

> "1. Generate possible Cb-Cf combinations for each possible set of reference assignments.
>
> 2. Filter by constraints, e.g., syntactic coreference constraints, selectional restrictions, centering rules and constraints.
>
> 3. Rank by transition orderings. "

A more computationally tractable algorithm is also presented in Brennan et al. (1987). In this algorithm, the preferred antecedents are computed from the relation between the forward and backward looking centers in adjacent sentences. Four intersentential relationships between U_n and U_{n+1} are defined which depend on the relationship between $C_b(U_{n+1})$, $C_b(U_n)$, and $C_p(U_{n+1})$. This is shown in Table (5.1):

Table 5.1: Transition rules for Centering algorithm

	$C_b(Un)=C_b(Un-1)$	**$C_b(Un) != C_b(Un-1)$**
$C_b(Un)=C_p(Un)$	Continuing	Smooth Shifting
$C_b(Un)!=C_p(Un)$	Retaining	Rough Shifting

As a very simple illustration, a simple resolution process with Centering approach is shown for the following discourse segment.

In this study, a new model is implemented based on the Centering Theory which tries to resolve pronominal pronouns in Turkish Language. The reason for choosing centering based algorithm is that some linguistic rules for Turkish which supports and provides the requirements of this theory are already discussed in former studies. Hence, this theorical studies are moved one step forward with the computational approach which is served in this study.

5.5 Other Approaches for Anaphora Resolution

In this section some of the other algorithms used in anaphora resolution process are summarized. Actually the algorithms and the theories served in this section does not form all of the research done in this branch. But those are discussed that have successfully tested and well-known.

5.5.1 Binding Theory

The binding theory is part of the principles and parameters theory (Chmosky 1981, 1995) and among other accomplishments, imposes important syntactic constraints as to how noun phrases may corefer. It accounts for the interpretation of anaphors including reflexive pronouns, personal pronouns and lexical noun phrases. The binding theory regards reflexives in English as short-distance anaphors and requires that reflexive anaphors refer to antecedents that are in a so-called local domain. Since reflexives are bound by their antecedents in this local domain, they are often called bound anaphors. In contrast, personal pronouns are free anaphors with respect to the same local domain, they are long-distance anaphors which permit antecedents to come only outside their local domain. Arriving at a useful definition of this local domain in structural terms has been an active area of research. As an illustration, consider the following examples:

(5.27)

Victoria believed George had seen herself

(5.28)

Victoria believed George had seen him.

In (5.27) the noun phrases Victoria and herself do not corefer because the reflexive is too far away: a reflexive pronoun must corefer with a noun phrase in the same local domain. On the other hand, in (5.28) George and him cannot corefer because they are too close: a non-reflexive pronoun cannot corefer with the noun phrase in the same local domain.

(5.29)

Sylvia believed she was the most diligent student

(5.30)

Sylvia believed he was the most diligent student

(5.31)

She believed Sylvia was the most diligent student.

In (5.29) Sylvia and she can be coreferential but in (5.30) coreference between Sylvia and he is not possible because the anaphor and the antecedent must agree in gender and number. In (5.31) coreference between she and Sylvia does not hold on this occasion one may be tempted to conclude that this is because the antecedent does not precede the anaphor. However, it is already told that in the case of cataphora (cf. Section 5.1), the anaphor may precede the antecedent:

(5.32)

[O_i] Eve geldiğinde, Ali_i çok yorgundu.

He home.DAT came.PAST.WHEN Ali very tired.3SG

As he was came home, Ali was very tired.

The explanation as to why in (5.31) no coreference is possible will be provided later by the constraint introduced in the section 5.5.1.3 of this study. The same constraint will also explain why in some cases coreference would be possible if a pronoun were used, as opposed to a lexical noun phrase such as “the young model” in the following example:

(5.33)

Ayşe genç modelin en güzel kız olduğunu düşünüyor.

Ayşe young model.GEN most beatiful be.PAST.POSS think.PROG

(Ayşe believes that the young model is the most beatiful girl.)

Before explaining these cases told in above, the structural relation of c-command should be introduced. Because this concept plays an important role in the constraints formulated in the next sections of the study.

A node A c-commands a node B if and only if A does not dominate B. B does not dominate B iff the first branching node dominating A also dominates B.

In figure (5.1) which illustrates the notion of c-command, it can be seen for example that:

B c-commands C and every node that C dominates

C c-commands B and every node that B dominates

D c-commands E and J, but not C, or any of the nodes that C dominates

H c-commands I and no other node.

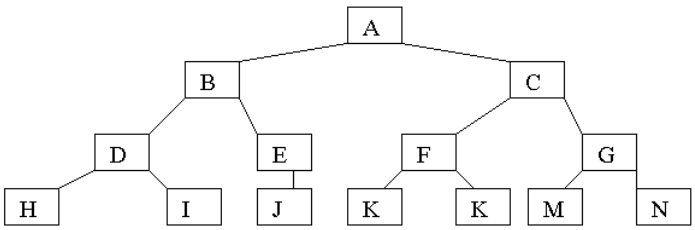


Figure 5.1: c-command illustration

5.5.1.1 Interpretation of Reflexives

The interpretation of reflexive anaphors is associated with factors such as grammatical agreement, c-command relation and local domain. To start, a reflexive anaphor must agree in person, gender and number with its antecedent. Another key constraint that delimits the interpretation of reflexives states that: A reflexive anaphor must be c-commanded by its antecedent.

A close examination of the examples and related figures are shown in following:

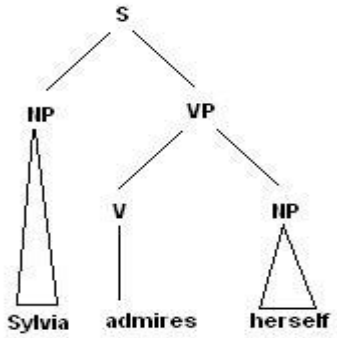


Figure 5.2: Sylvia admires herself

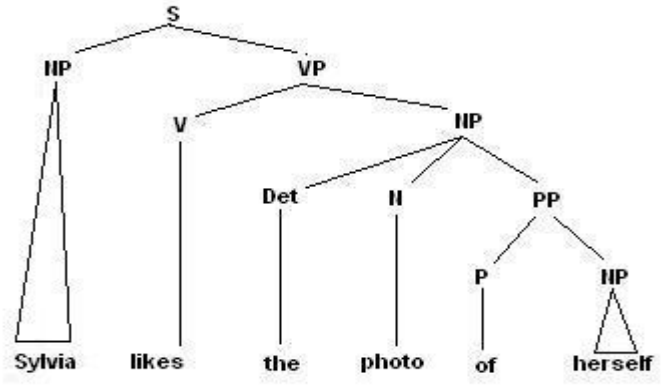


Figure 5.3: Sylvia likes the photo of herself

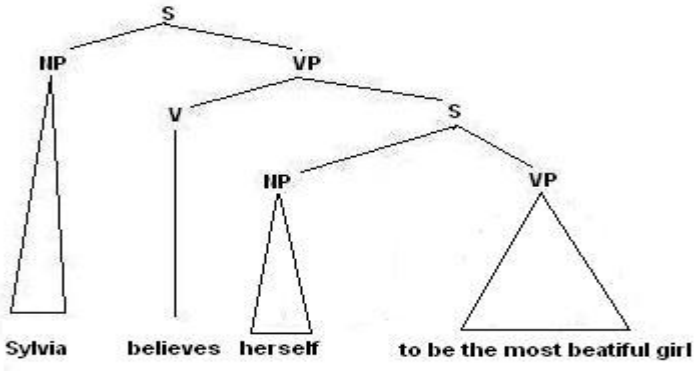


Figure 5.4: Sylvia believes herself to be the most beatiful girl.

In these examples “herself” is c-commanded by Sylvia.

5.5.1.2 Interpretation of Personal Pronouns

The interpretation of non-reflexive pronominal anaphors differs from that of reflexives. From the examples:

(5.34)

Sylvia admires herself

(5.35)

Sylvia admires her.

it is clear that whereas herself is bound and refers to Sylvia, the pronoun her, which is in the same syntactic position as herself, is free within the domain defined by the sentence must refer to an entity different from Sylvia and outside this domain. Note that Sylvia c-commands both herself and her.

The domain in which pronominal anaphors are free is the same as the domain in which the reflexives are bound (Haegamen, 1994). The antecedent of a reflexive lies within the local domain of the reflexive anaphor and c-commands it. On the other hand a noun phrase and the pronominal anaphor cannot be coreferential if the noun phrase is situated in the local domain of the anaphor and c-commands it. The main constraint in the interpretation of pronouns stipulates that: A pronoun cannot refer to a c-commanding NP within the same local domain.

This constraint has been used in automatic anaphora resolution to narrow down the search scope of candidates for antecedents. The details of the subject can be found in Ingria and Stallard (1989).

5.5.1.3 Interpretation of Lexical Noun Phrases

Lexical noun phrases are the class of noun phrases which are not pronouns, such as “Ayşe” or the “young model” in example (5.33). These types of noun phrases, also referred to as “referential expressions”, are inherently referential, select their reference from the universe of discourse and therefore have independent reference. In contrast to reflexive pronouns which must be bound locally, or non-reflexive pronouns which must be free locally but may be bound outside their local domain, referential expressions must be free everywhere, that is, they cannot bund by an antecedent within and outside their local domain.

An important constraint delimiting the interpretation of lexical noun phrases states that: A non-pronominal NP cannot corefer with an NP that c-commands it.

This constraint has been used in anaphora resolution systems to discount coreference in examples such as:

(5.36)

She admires Sylvia

(5.37)

She likes a photograph of Sylvia

(5.38)

Sylvia said the young model was the most beatiful girl.

In these examples the non-pronominal noun phrases and the young model are c-commanded by the NPs She and Sylvia respectively and, therefore, cannot be coreferential with them.

The binding theory is helpful in determining impossible antecedents of pronominal anaphors and in assigning possible antecedents to bound anaphors, and some of the constarints outlined above have been used for automatic anaphora resolution in the studies; Ingria and Stallard (1989) and Carvalho (1996).

However, the theory is still an active area of research in syntax and is not yet fully developed: there are still a number of cases that cannot be accounted for.

5.5.2 RAP Algorithm for Anaphora Resolution

Lappin and Leass (1994) provides an algorithm for pronoun detection. The algorithm employs a simple weighting scheme that integrates the effects of recency and syntatically-based preferences; no semantic preferences are employed beyond those enforced by the agreement.

The algorithm termed Resolution of Anaphora Procedure (RAP) operates on syntactic representations generated by McCord's Slot Grammar parser (McCord 1990, 1993). It relies on salience measures derived from the syntactic structure as well as on a simple dynamic model of attentional state to select the antecedent of a pronoun from a list of NP candidates. It does not employ a semantic information or real-world knowledge in choosing from the candidates. RAP contains the following main components:

The anaphor binding algorithm uses the following hierarchy of argument slots:

subj > agent > object > iobj > pobj **(5.1)**

Here subject is the surface subject slot as identified by the slot grammar parser, agent is the deep subject slot of a verb heading a passive VP, obj is the direct object

slot, iobj is the indirect object slot and pobj is the object of a PP complement of a verb, as in put NP on NP. A noun phrase NP is the antecedent for a reflexive or reciprocal pronoun R iff R and NP do not have incompatible agreement features, and any of the following conditions hold:

a) R is in the argument domain of NP, and NP fills a higher argument slot than R.

(5.39)

They$_i$ wanted to see themselves$_i$.

(5.40)

Mary$_i$ knows the people who John introduced to each other$_i$.

b) R is in the adjunct domain of NP:

(5.41)

He$_i$ worked by himself$_i$

(5.42)

Which friends$_i$ plan to travel each other$_i$.

c) NP is an argument of a verb V, there is a noun phrase Q in the argument domain or the adjunct domain of NP such that R has no noun determiner, and R is an argument of Q, or an argument of preposition PREP and PREP is an adjunct of Q.

(5.43)

They$_i$ told stories about themselves$_i$.

d) R is determiner of a noun Q, and Q is in the argument domain of NP and NP fills a higher argument slot than Q, or Q is in the adjunct domain of NP.

(5.44)

[John and Mary]$_i$ like each others$_i$ portrait.

e) R is in the noun phrase domain of NP

(5.45)

John likes Bill's$_i$ portrait of himself$_i$.

Salience weighting applies to discourse referents and is computed on the basis of salience factors. In addition to sentence recency (where recent sentences are given higher weight), the algorithm gives additional weight to subjects (subject emphasis), predicate nominals in existential constructions (existential emphasis), direct objects (accusative emphasis), noun phrases that are not contained in other noun phrases (head noun emphasis) and noun phrases that are not contained adverbial prepositiponal phrases (non-adverbial emphasis). The salience factors and their weights for English are given in Table (5.2):

Table 5.2: Salience factor types with initial weights

Factor Type	**Initial Weight**
Sentence recency	100
Subject Emphasis	80
Existential Emphasis	70
Accusative Emphasis	50
Indirect Object and Oblique Complement Emphasis	40
Head Noun Emphasis	80
Non-adverbial Emphasis	50

The RAP's procedure for identifying antecedents of pronouns works as follows:

a) First a list of all NPs in the current sentence is created and the NPs are classified according to their type (definite NP, pleonastic pronoun, other pronoun, indefinite NP).

b) All NPs occuring in the current sentence are examined.

c) NPs that evoke new discourse referents are distinguished from NPs that arepresumably coreferential with already listed discourse referents as well as from those used non-referentially (e.g pleonastic pronouns)

d) Salience factors are applied to the discourse referents evoked in the previous steps as appropriate.

e) The syntactic filter and reflexive binding algorithm are applied.

f) If the current sentence contains any personal or possesive pronouns, a list of pronoun-NP pairs from the sentence is generated. Th pairs for which coreference is ruled out on syntactic grounds are identified.

g) If the current sentence contains any reciprocal or reflexive pronouns, a list of pronoun-NP pairs is generated so that each pronoun is paired with all its possible antecedent binders.

h) If any non-pleonastic pronouns are present in the current sentence, their resolution is attempted in the linear order of pronoun occurence in the sentence.

i) In the case of reflexive or reciprocal pronouns, the possible antecedent binders are identified by the anaphor binding algorithm. If more than one candidate is found, the one with the highest salience weight is chosen.

j) In the case of third person pronouns, a list of possible antecedent candidates is created. It contains the most recent referent of each equivalence class. The salience weight of each candidate is calculated as the sum of the values of all salience factors that apply to it, and included in the list. The salience weight of these candidates can be additionally modified. Also, the salience weights of candidates from previous sentences are degraded by a factor of 2 when each new sentence is processed. Unlike the salience factors shown in Table (5.2), these modifications of the salience weights are local to the resolution of a particular pronoun. Next, a salience threshold is applied: only those candidates whose salience weight is above the threshold are considered further.

k) In the final step agreement of number and gender is checked. This procedure seems to be much simpler for English than for Turkish and other languages, which may exhibit ambiguity of the pronominal forms as to gender or number. First the morphological filer is applied, followed by the syntactic filter. If more than one candidate remains, the candidate with the highest salience weight is chosen. In the event of more than one candidates remaining, the candidate closest to the anaphor is selected as he antecedent.

For more details of the stages of the algorithm, see Lappin and Leass (1994).

The blind test was performed on 360 pronoun occurences, which were randomly selected from a corpus of computer manuals containing 1.25 million words. RAP performs succesfull resolution in %86 of the cases, with %72 sucess for the intersentential cases (alltogether 70) and %89 for intersentential cases (alltogether 290).

5.5.3 Hobbs's Approach for Anaphora Resolution

Hobbs (1976, 1978) proposed two approaches to pronoun resolution; one syntactic operating on syntactic trees and another using semantic knowledge. In this section of the study, syntactic treatment of his will be focused.

Hobbs's algorithm operates on surface parse trees and on the assumption that these represent the correst grammatical structure of the sentence with all adjunct phrases properly attached, and that they feature syntactically recoverable omitted elements such as elided verb phrases and other types of zero anaphors or zero antecedents. Hobbs also assumes that an NP node has an N-bar node below it, with N-bar denoting a noun phrase without its determiner. Truly adjunctive prepositional phrases are attached to the NP node. This assumption, according to Hobbs, is necessary to distinguish between the following two sentences:

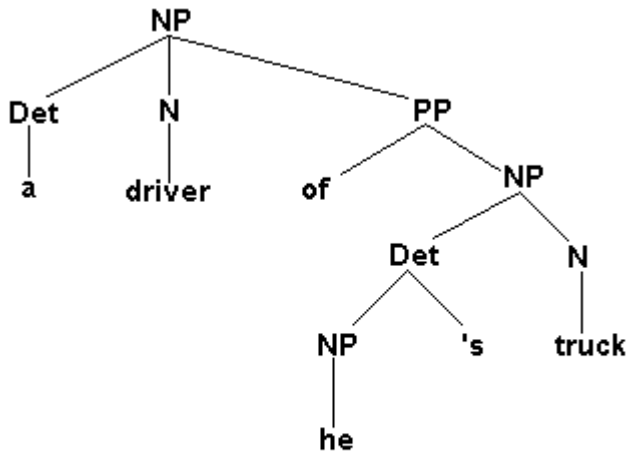
(5.46)

Mr. Smith saw a driver in his truck.

(5.47)

Mr. Smith saw a driver of his truck.

In (5.46) his may refer to the driver, but in (5.47) it may not. The structures to be assumed for the relevant noun phrases in "a" and "b" are shown in Figure (5.5):



a)

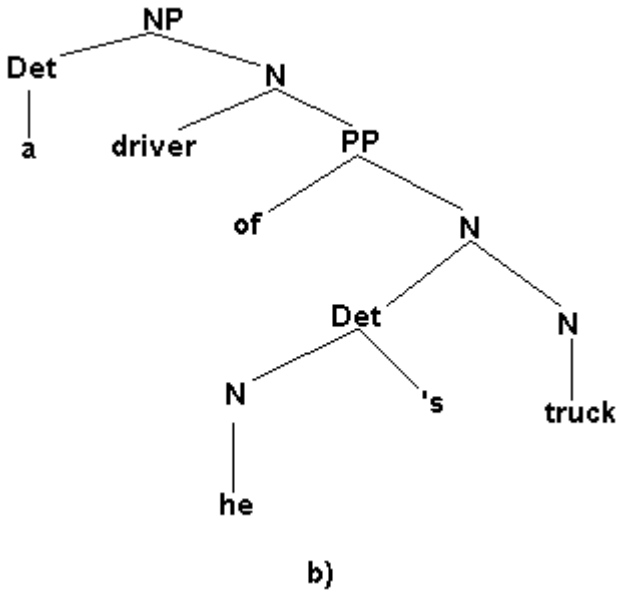


Figure 5.5: Syntactic structures corresponding to (5.39) and (5.40)

Hobbs's algorithm traverses the surface parse tree in a particular order looking for a noun phrase of the correct gender and number. The traversal order is detailed as the following (Hobbs 1976, 1978):

"1. Begin at the NP node immediately dominating the pronoun in the parse tree of the sentence S.

2. Go up the tree to the first NP or S node encoutered. Call this node X, and call the path used to reach it "p".

3. Traverse all branches below node X to the left of path "p" in a left-to-right, breadth-first fashion. Propose as the antecedent any NP node encountered that has an NP or S node between it and X.

4. If thenode X is the highest S node in the semtence, traverse the surface parse trees of previous sentences in the text in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node encountered, it is proposed as antecedent. If X is not the highest node in the sentence, proceed to step 5.

5. From node X, go up to the tree to the first NP or S node encountered. Call this node X and call the path traversed to reach it "p".

6. If X is an NP node and if the path "p" to X did not pass through the N-bar node that X immediately dominates, propose X as the antecedent.

7. Traverse all branches below the node X to the left of path "p" in a left-to-right, breadth-first manner. Propose any NP node encountered as the antecedent.

8. If X is S node, traverse all branches of node X to the right of path "p" in a left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as the antecedent.

9. Go to step 4."

Steps 2 and 3 of the algorithm take care of the level in the tree where a reflexive pronoun would be used. Steps 5-9 cycle up the tree through S and NP nodes. Step 4 searches the previous sentences in the text.

Hobbs evaluated his algorithm on 300 pronouns from three different texts: 100 of these pronouns were from William Watson's Early Civilization in China, 100 were from the first chapter of Arthur Haley's novel Wheels and 100 from the 7 July 1975 edition of Newsweek. The pronouns were he, she, it and they; it was not counted when referring to a syntactically recoverable "that" clause or when pleonastic.

Hobbs concluded that whether the success rate was %92, %91.7, %81.8, the results showed that the naive approach was very good. In its original form, Hobbs's algorithm was simulated manually. As a consequence, it operated on perfectly analysed sentences and the success rates of %88.3 and %91.7 given by Hobbs should be regarded as ideal.

Besides centering and the other algorithms described above, there are several other techniques for anaphora resolution, intended to deal with especially English text fragments. It is founded it worth briefly mentioning them before proceeding on to the presentation of our own account. Givon (1976) proposes a universal hierarchy of topicality. According to him, this hierarchy can be expressed as one linear expression, such as "Subject > Definite Object > Human Object > Indefinite" Givon (1983), constructs a topic continuity device to account for how referential expressions are used in the discourse. Ariel (1988), presents an approach which is similar to Givon (1983)'s. He does not take care of the hierarchical structure of the discourse or differentiation of discourse referents with respect to various criteria. Fox (1987) underlines the relationship between anaphor interpretation and discourse structure. Kuno (1989) deals with zero pronouns in Japanese discourse. McCord (1990) presents algorithms for handling three different sorts of anaphora.

6. Anaphora Resolution For Turkish

In previous chapters of this study, very general terminology and the overall requirements of the anaphora resolution is given. Since such a procedure is dealed with any natural language, beside the computational algorithms that are necessary for aoutomatic resolution, the structure and rules of the language is also very important. Actually, the whole operation is based on these language rules.

In this study, an anaphora resolution model is presented and this model is implemented for Turkish based on the Centering Theory rules. Centering theory, however, does not propose an analysis for evoking discourse entities which enable them to be referred to anaphorically by definite pronouns and full NPs, but rather is conserned with tracking anaphors once they are evoked in the discourse model. So, before starting the detailed discussion of the model we have to concern on the form and content of the NPs that first evoke entities in the discourse model.

6.1 Anaphors and Pronouns In Turkish

In this section of the study, some definitions and findings related with the most basic concept, anaphor, is observed. This observation is made especially for Turkish language since this is study based on this language.

Turkish is pro drop language with null subjects that are locally identified via agreement morphmemes on verbs. Yüksel (1997) claims that English anaphors and pronouns have gender, person, and number features, while Turkish anaphors and pronouns lack the gender feature.

As discussed both in Turanlı (1996) and Yüksel (1997), there some null pronouns and anaphors which are the dropped version of the overt pronouns and anaphors.

In Yüksel (1997), it said that Languages like English do not allow the drop of pronouns, but in a pro-drop language like Turkish, they can be dropped under certain conditions.

Consider the following example:

(6.1)

Ahmet$_i$ arabasıyla$_k$ çabucak eve vardı.

Ahmet car.WITH quickly home.DAT arrive.PAST.3SG

Ahmet arrived home with his car quickly.

(6.2)

[O$_i$] Onu$_k$ garaja park etti.

It.ACC garage.DAT park.PASS.3SG.

He parked it into the garage.

In (6.2) the anaphor “O” is dropped and became zero (or null). Null and overt subjects in Turkish occur both in main and subordinate clauses, and both necessarily agree with the number and person morphology on the verb. Third-person singular agreement morpheme is null and third-person plural morpheme is optional. As said before, gender is not marked in Turkish.

As stated in Yüksel (1997), anaphors are seperated into two groups, namely reflexives and recipocals both in English and Turkish. Turkish anaphor and pronoun type are shown in Table 6.1 below:

Table 6.1: Turkish Anaphors and Pronouns.

Anaphors		Pronouns
Reciprocals	Reflexives	O Onu Onun Onlar Onları Onların ϕ (null)
birbirleri -Iş	Kendi Kendim Kendin Kendimiz Kendiniz ϕ (null)	

In this table, “-Iş” is the suffix indicating a morphological reciprocal where “I” denotes an underspecified high-vowel, its realization as ı/i/u/ü is determined by vowel harmony.

These pronouns shown in the Table 6.1 may have a zero nominative case in the subject position. They can also overtly assigned accusative, dative, ablative in the object position depending on a structural and inherent case-assigning verb.

6.1.1 Null Subjects in Turkish

As claimed in Turanlı (1996), “null subjects” can occur in the following contexts. Examples are taken to give the case birefly:

a) Simple Clauses

Null subjects can occur in simple clauses:

(6.3)

[Ben] Dün bir kitap aldım.

yesterday one book buy.PAST

(I) bought a book yesterday

b) Subordinating Clauses

The subject of subordinate clause may be null in some cases in Turkish:

(6.4)

$Orhan_i$ [[O_i] çalışırken] müzik dinler.

Orhan work.WHEN music listen.AOR

Orhan listens to music when (he) works

Sometimes the subject of subordinate clasuse may not be null also. This discussion is done in Turanlı (1996) in detail.

c) Possesive NP’s

Turkish has the following possesive pronouns and they agree in number and person with the possessed noun as N-GEN N-POSS. In the examples below, in all these cases the subject of the possessive NP can be null:

(6.5)

benim evim.

I.GEN.1SG house.POSS.1SG

(my house)

(6.6)

senin evim.

you.GEN.2SG house.POSS.2SG

(your house)

(6.7)

onun evi.

he/she/it.GEN.3SG house.POSS.3SG

(his/her/its house)

(6.8)

bizim evimiz.

we.GEN.1PL house.POSS.1PL

(our house)

(6.9)

sizin eviniz.

you.GEN.2PL house.POSS.2PL

(your house)

(6.10)

onların evi.

they.GEN.3PL house.POSS.3PL

(their house)

d) Expletive Pronouns

Turkish does not have expletive (ambient) pronouns as in the weather verbs:

(6.11)

Yağmur yağıyor

rain rain.PROG

([It] is raining)

6.1.2 Null Objects in Turkish

There are also some cases the null objects may occur in Turkish language. In the following, there are some examples that null objects occur in Turkish.

(6.12)

Gazeteyi$_i$ gördün mü?

Newspaper.ACC see.PAST.2SG Quest

(Have you seen the paper?)

(6.13)

[Onu$_i$] [Ben] Görmedim.

see.NEG.PAST.1SG

([I] didn't see [it])

And also there are some other null pronouns that are not in the subject or the object position. In the example below "onun" (it) is in genitive case and it is in adjunct position:

(6.14)

Bahçedeki ağaç$_i$ çiçek açtı.

Garden.LOC tree bloom.PAST

(The tree in the garden bloomed.)

(6.15)

[Biz] [Onun$_i$] Altında oturduk.

beneath.POSS.3SG.LOC sit.PAST.1PLU

([We] sat beneath [it])

6.2 Referential Expressions For Turkish

In this chapter the referentiability for Turkish is discussed. Actually this disccussion is well made in Turanlı (1996). In that study, many other related research for Turkish on that subject is birefly explained and findings are served. The main purpose of Turanlı's study is to determine the cases that overt or null pronouns will be used. Since this study is actually based on Turanlı (1996) and additionally is an implementaiton of that, these linguistic findings are briefly taken from that research.

As inferred from the findings, in general overt pronouns are more suitable in the cases where the antecedent is a referential expression. As discussed in section 6.3, the scope of this study involves with some some limited pronoun types in Turkish, which are "O", "Onlar", "Onun", "Onların" as overt or null only if they are in the subject position. For this reason, only the referential expression cases are taken from the same study since other pronouns such as null in the position that is not subject require non referential expression cases.

In Turanlı (1996), a referential expression is described as an NP that evokes a discourse entity in a discourse model. A non-referential expression does not create a new file card or evoke a discourse entity, and cannot be discussed anaphorically by definite reference in the subsequent discourse. As said in that study, it is impossible to ask a question like "Which X is E?" where E is the non-referential expression.

In the following parts of this section referential and nonreferential contexts and their roles evoking discourse entities will be discussed.

An "indefinite predicative nominal" will not evoke a discourse entity (Kuno 1970, Webber 1979 and Appelt 1985). Consider the following example:

(6.16)

Her şey sonsuz bir yarış$_i$.

Every thing endless one competiton.

(Everything is an endless competition)

(6.17)

Ben sevmem [#onları$_i$] yarışları.

I like.NEG.1SG competition.PLU.ACC

(I don’t like competitions.)

Since “yarış” is predicative nominal in (6.16) it cannot be accessed by using the pronoun “onları” in (6.16). Because it does not evoke a discourse entity.

As stated in Prince (1981), a discourse entity can either be textually evoked, or it can be situationally evoked by pointing at some entity in the perceptual field. If a cat enters the room, its appearence makes it a salient entity and enables the speakers to refer it. Likewise, in the following example “adam” is used with a demonstrative and it also defines a situationally evoked entity:

(6.17)

Şu adam bize bakıyor.

That man we.ACC look.PROG.

That man is looking at us

According to Dede (1986) and Erguvanlı (1984) “accusative case” defines definiteness and referentiality in general.

(6.18)

Ali_i $araba\imath_k$ sürüyor.

Ali car.ACC drive.PROG.3SG

Ali is driving the car.

(6.19)

$[O_i]$ Onu_k iyi kullanıyor.

İt.ACC good use.PROG.3SG

([He] is using it good.)

Furthermore, Dede (1986) shows that some NPs in “generic sentences” are optionally accusative case-marked in the object position in Turkish. Erguvanlı (1984) proposes that animacy plays an important role in the assignment of accusative case in generic sentences. As a result of these two proposals, consider the examples below:

(6.20)

Çocuklar çukulata sever.

Child.PLU chocolate like.AOR.

(Children like chocolate.)

(6.21)

Çocuklar çukulatayı sever.

Child.PLU chocolate.ACC like.AOR.

(Children like chocolate.)

As shown in the previous example accusative case is optionally used. In these sentences “çukulata” is used as generic and it is inanimate. In the following example, however, insanlar is also generic but it animate. So accusative case-marked usage is a must in this case:

(6.22)

Ben insanları severim.

I human.PLU.ACC like.AOR.1SG

(I like human beings.)

As a result, it appears that accusative case has a function to declare a unique entity in the discourse.

As stated in Turanlı (1996), indefinite nonspecific NPs within the scope of a negative operator cannot evoke discourse entities. Consider the example given in that study:

(6.23)

Şemsiyem yok.

Umbrella not exist

(I don’t have an umbrella)

(6.24)

Ben sana bulurum.

I you.DAT find.PRES.1SG.

(I will find you one.)

In (6.23) above, “şemsiye” is a nonreferential expression so an overt pronoun is prohibited in (6.24). However, it is also possible to evoke an entity in negotiation sentences by using an adverb like “artık” (any more). Consider the example below:

(6.25)

Artık Ali’nin evi_i yok.

Any more Ali.GEN house.POSS.3SG not exist.

(Ali doesn’t have a house any more.)

(6.26)

O_i geçen hafta satıldı.

It last week sell.PASS.PAST

(It was sold last week)

Turanlı (1996) pointed out that in definite nonspecific NPs with in the scope of yes/no questions do not evoke entities:

(6.27)

Ali_k yeni bir $araba_i$ mı aldı?

Ali new one car QUEST buy.PAST.

(Did Ali buy a new car?)

(6.28)

Hayır evini sattıktan sonra #onu_i alacak. /O_k^*

No house.POSS.3SG.ACC sel.NOM.ABL after it.ACC buy.FUT.

(No, he will buy one when (he) sells his house)

In (6.27) above, araba is nonreferential expression. So it cannot be referenced by an overt pronoun in (6.28).

It has been stated in Turanlı (1996) that definite direct objects in Turkish are overtly accusative case-marked, while indefinite objects do not have overt case markings. Indefinite objects are modified by numeral bir (one) and indefinite quantifiers bazı, biraz (some), bir kaç (a few) and these objects are not overtly accusative case-marked unless they are specific (Enç, 1991). There is another set of objects which are neither overtly accusative case-marked nor modified by the indefinite numeral or any other modifier. These objects are referred to as “bare objects”.

It is shown in Turanlı (1996) that objects that are not case-marked behave differently than those that are case-marked. Objects that are not overtly case-marked cannot occur in sentence initial or postverbal positions, but are strictly anchored to the preverbal slot. They cannot be seperated from the governing verb by an NP by an adverbial. In (5.29) while the accusative case-marked kitabı (book.ACC) can be seperated form the verb by an adverb, the bare object should be anchored to the preverbal position.

(6.29)

Ali kitabı yavaş yavaş okuyor.

Ali book.ACC slow slow read.PROG

(Ali is reading the book slowly)

(6.30)

Ali #kitap yavaş yavaş okuyor.

(6.31)

Ali yavaş yavaş kitap okuyor.

Ali slow slow book read.PROG

(Ali is reading some book slowly)

The immobility of bare objects has led some researchers to argue that they undergo a process Noun Incorporation (NI). As a detailed invastigatin done in Turanlı (1996), noun incorporation is optional. As a result, if incorporation applies, the bare objects in Turkish cannot behave as discrete entities and independent constituents. Cnsider the example below:

(6.32)

Çağlar oyun oynuyor.

Çağlar game play.PROG

(Çağlar is playing the game) or (Çağlar is playing some game.)

The object oyun (game) in (6.32) will either be incorporated or a discerete constituent. If it is incorporated, game will not evoke a file card, it it is not, however, game will be discrete entity.

6.3 Scope of This Study

The talents of the model suggested in this study is served in the current section. Because of the different nature of pronouns for every different language, the resolution process will also be different for every other language. For example, the pleonastic “it” which exist in English, does not exist in Turkish. So such cases cannot be handled in the resolution process of this study. In following the ability of this study is presented in detail.

In general, definite noun phrases are potentially anaphoric often referring back to preceeding noun phrases, as the “kraliçe” in the following example:

(6.33)

Kraliçe Elizabeth törene katıldı.

Queen Elizabetn ceremony.DAT atten.PAST.3SG.

(Queen Elizabeth attended the ceremony.)

(6.34)

Kraliçe bir konuşma yaptı.

The queen one speech do.PAST.3SG

(The queen delivered a speech.)

It is important to bear in mind that not every definite noun phrase is necessarily anaphoric. In (6.36) the NP bakan (the minister) is not anaphoric and does not refer to başkan (the president) in (6.35):

(6.35)

Başkan toplantıya katıldı.

President meeting.DAT attend.PAST.3SG.

(The president attended the meeting)

(6.36)

Bakan da oradaydı.

Minister too there.PAST.3SG.

(the minister was there, too.)

In this study, such definite pronouns are not resolved. Actually, it is important for an anaphora resolution program to be able to identify those definite descriptions that are or are not anaphoric. Fortunately, this computational model is implemented with an object oriented (modular) approach which means it is very easy to add such an external module to handle these NP's in later research.

As discussed in former sections, anaphora resolution process needs the input sentences that are morphologically labeled and syntactically parsed by some morpholical analyzer and POS tagger applications. Such applications are not in the scope of this study. But the input information needed for the resolution process are provided by Oflazer (1994) and Oflazer et. al (2002). Since the reference provides only the morphologically analyzed information, the necessary syntactic labeling (which gives the sentence role information of any word such as: subject, adjunct, object and etc.) are completed manually for this study.

Genericity (cf. Section 6.2) is another issue awaiting for an explanation in terms of how it interacts with anaphora resolution. As should have been noted, this level of interpretation is ignored in this study. Also, it is needless to say that more semantic, pragmatic or syntactic constraints need to be incorporated to the system in order to generate more accurate results.

As explained in section 5.3.2, search scope also an important issue in anaphora resolution systems. Definite noun phrases can refer further back in the text and for such anaphors search scope is normally longer than 1, 2 or 3 sentence before. In this study, the search scope is assumed as only one sentence back. Actually this assumption is made because of the nature of the Centering Theory which provides that the centers are always moved with the following sentence.

As a result, this study involves with the third person pronominal pronouns in all cases, null pronouns only in the subject position of the sentence and the pronouns which point to a location. They can be listed as; o, onu, ona, onda, ondan, onun,

onlar, onları, onlara, onlarda, onlardan, onların, ora, orayı, oraya, orada, oradan, oranın. The level of the input sentences of the model which tries to resolve anaphors in Turkish language is at the primary school level. Hence, the suggested hierarchical rules and success of the computational model is tested by using this simple structured sentences as the beginning.

7. Computational Model For Anaphora Resolution in Turkish

In this study a computational model for anaphora resolution in Turkish that is based on Centering Theory is presented. The model is primarily based on the linguistic analysis, where Centering Theory is applied to anaphoric phenomena in Turkish. This study's aim is twofold. One of the objectives is to provide an effective implementation of the centering approach for Turkish. But also, it is aimed to detect some defective aspects of this approach that become highlighted from a computational point of view.

And also, a new hierarchy is suggested in this study. This hierarchy is used in ranking process that is required for Centering Algorithm. Since the former studies are developed for mostly in English, the hierarchies that are used for this language as the salience criteria do not produce the true results for Turkish. For this reason, it is tried to implement a new computational model for anaphora resolution in Turkish by taking the rules related for the salience of referential expressions in Turkish proposed mostly in Turanlı (1996). After testing the accuracy of this new hierarchy, it is included and used in the application.

In this section, the suggested computational model and all of the modules involves it is explained in detail. The algorithms and the theories that the model is based on are already explained in previous sections.

Actually all segments of the main application is not completely implemented yet. Anyway, a real model required for the anaphora resolution is designed firstly and many parts of the segments are implemented to observe the results of the study.

In this section, an architectural description of designed system is introduced. The system is constituted by three sub-systems: POS Tagger, Parser, Anaphora Resolvent.

When a sentence comes into the system, all strings of the sentence will be assigned appropriate parts of speech (POS) tags by the POS Tagger. As discussed in former sections, the operation gives the case, time, gender and number informations for any word in given sentences. Without this step, anaphora resolution system can not be implemented since this criteria is needed for eleminating process in preceding modules.

Then, these tagged input is sent to a parser application. The Parser specifies the position of these strings in the sentence. The sentence position information is also very important for the model since the hierarchy constraints includes the position information.

These two processes gives the input a format required by the Anaphora Resolvent. This third module is responsible for finding a unique antecedent for a given anaphor. A diagrammatic representation of the whole process is shown in Figure 6.1:

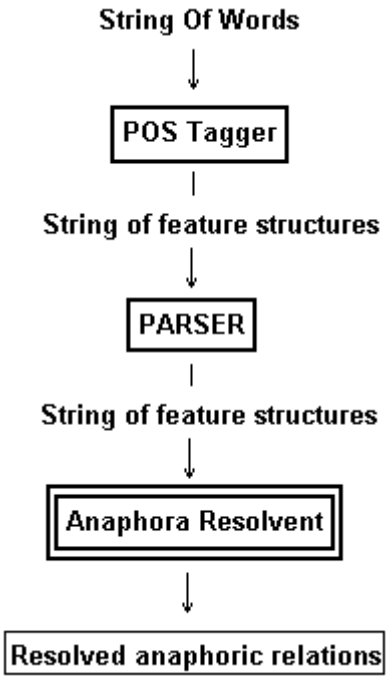


Figure 7.1: Overall Model for Anaphora Resolution System

As discussed in previous sections, implementing POS tagger and parser applications needed for this model is not in the scope of the study. Instead, the morphological analyzer is used for POS tagging which is firstly developed as told in Oflazer (1994). And also, the parse operation of the data is completed manually.

In the following section, all of the modules are explained in detail.

7.1 Low Level Design

In this section, we will go into the details of the “Anaphora Resolvent”, which is the most significant module in terms of what we contribute to the algorithmic analysis of anaphora resolution in Turkish. Actually this main module is also implemented as modular as will be discussed in following. As a result, the anaphora resolving operation in general, is presented as an application of which every step is processed by another submodule located in it. By this way, it will be very easy to change the

operation of an existing module or to add a new one inside the apllication in further studies. And also, some of the modules presented in the low-level design will not prevent the working of the main application even if they are not implemented yet.

The main part forms the Resolvent part is shown in figure 7.2.

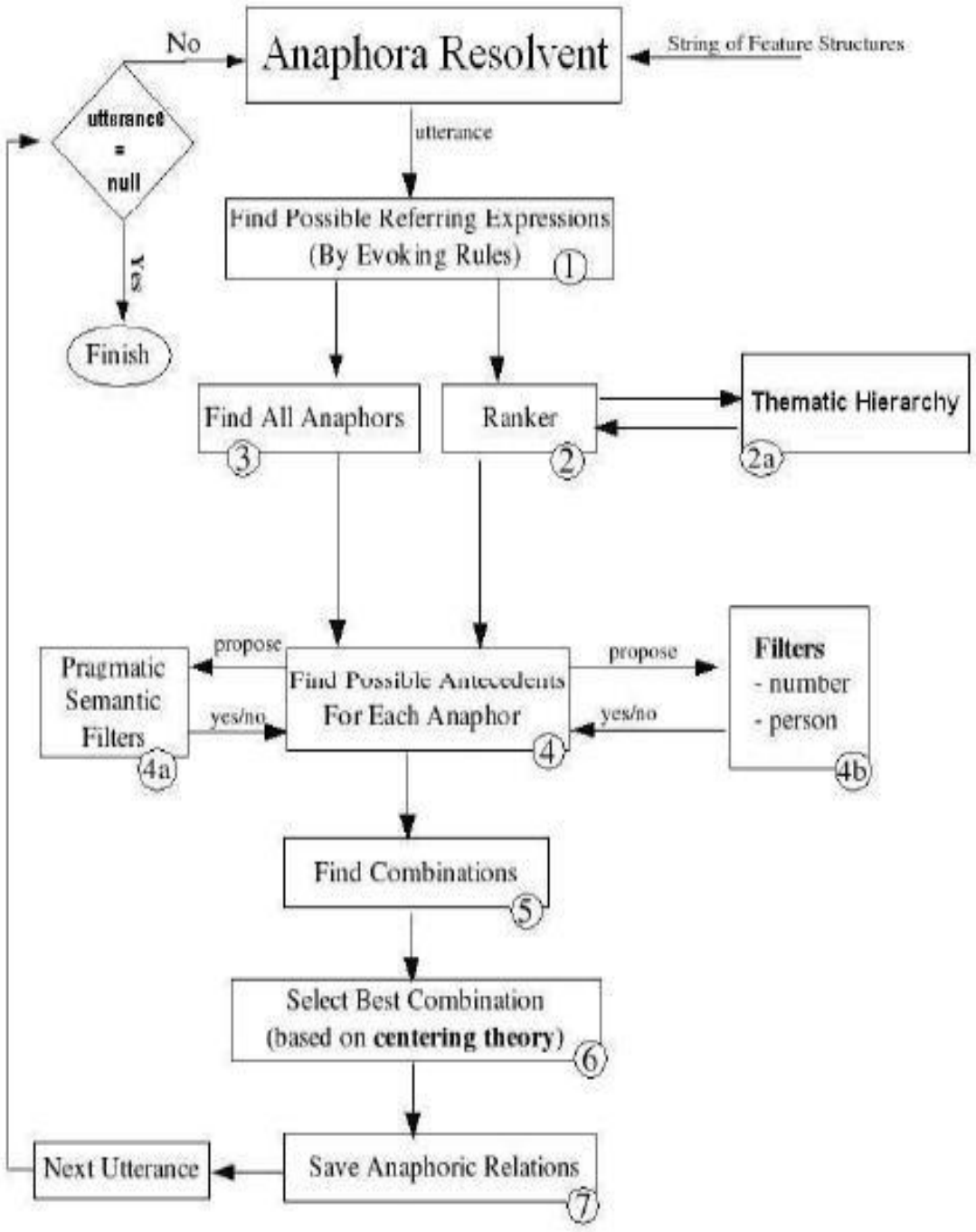


Figure 7.2: Low Level Design for Anaphora Resolvent

The subparts or submodules included in Anaphora Resolvent are explained seperately in below.

7.1.1 Module: Find Possible Referring Expressions:

Finding possible referring expressions is an important step to decide whether an element in the utterance is a referring expression or not. If an expression does not evoke an entity, it can not be referenced by an anaphor in the following utterances. Thus, in such cases the element cannot be a possible antecedent.

Actually the operation will also be different according the structure of the language. Because criterias of forming or not forming the referential expression are differs. The suitable criterias for the structure of Turkish language are selected for the study. Actually the rules adopted in section 5 and especially in section 5.2, are explained in detail. As explained in there, all of the assumptions are taken from Turanlı (1996) and from the others referenced in the same study. Anyway, the criterias sholud be listed briefly in below:

a) An indefinite predicative nominal will not evoke a discourse entity.

b) An indefinite predicative nominal will not evoke a discourse entity.

c) Accusative case marking evokes a unique entity in general.

d) If a generic noun is used in accusative case, it can be reached by a pronoun. But it cannot be reached if it is in nonaccusative case and used with an adjective.

e) In definite nonspecific NP's within the scope of negative operator cannot evoke discourse entities.

This module takes any sentence whose all words are morphologically analyzed and parsed, as the input and gives a list of words which can be assumed as the referring expression as the output. In the following example, if the sentence is given to this module:

(7.1)

Ali bügün okula gidecek.

Ali today school.DAT go.FUT.3SG.

(Ali will go to the school today.)

The output will be the following list: {Ali, okul} since Ali (proper name: Ali) and okul (school) can be taken as the referring expression for Turkish language according the rules discussed in the sections above.

7.1.2 Module: Ranker

The ranking process is a vital task for a Centering approach. Since the centering approach requires some Cp and Cb values, we need a hierarchy that order utterance elements according to their salience.

In this study, a new hierarchy is developed based on the prievious studies. This development process is explained in section 8 with the former experiments and results.

To sum up, this module uses the hierarchy suggested in related section to define the ordered version of the C_fList of the sentence which is taken as the input. And this order list will be sent as the output.

7.1.3 Module: Find All Pronouns

A prerequisite for anaphora resolution is, of course, to have some anaphors whose antecedents to be found. Some of the referring expressions may be pronoun in currently taken sentence. And also, the anaphoric relations may not be defined at that time. Hence such a module used to define the list of these unresolved pronouns is needed.

In the example below, the referring expressions given in the first sentence are matched with some pronouns in second one. But it should be note that, at the time that the second sentence is taken, the reolution is not operated yet.

(7.2)

Ali_i $okula_k$ gidecek.

Ali school.DAT go.FUT.3SG.

(Ali will go to the school)

(7.3)

Ama [O_i] $oraya_k$ araba ile gitmek istiyor.

But [he] there.DAT car with go want.PROG.3SG.

(But [he] wants to go there by car.)

As shown in figure below, this module takes the list created by the Find Possible Referring Expressions module as the input and outputs a pronoun list that the anaphoric relations are not defined.

Figure 7.3: Output list of module “Find All Pronouns”

This module will understand for anyword that is pronoun or not by taking the morphological label. And also, the knowledge representation used in the study records the necessary information of the reference word of any pronoun if it exists. By the help of these informations module can easiky define the entities that are not matched with any other antecedent.

7.1.4 Module: Find Possible Antecedents For Each Anaphor

Given an anaphor, it is possible to eliminate some of the antecedents that do not agree with this anaphor in terms of some linguistic criteria. To this effect, several filters that rule out certain anaphora relations on the basis of syntactic and semantic constraints have deveoped. Two constraints that play a part in our filtering process are that an anaphor and its possible antecedent must display a number-person agreement and that they are not likely to appear within the same sentence.

To give an idea, in the following example, the subject of the first sentence cannot be a possible antecedent of the pronominal subject of the second sentence, due to failure of number agreement.

(7.4)

Arkadaşlarım$_k$ beni$_i$ Ali$_j$ ile tanıştırdılar.

Friend.PL.POSS.1SG I.ACC Ali with introduce.PAST.3PL.

(My friends$_k$ introduced me$_i$ to Ali$_j$.)

(7.5)

O$_j$ çok iyi bir çocuğa benziyordu.

he/she/it very nice one boy.DAT look.like.PROG.PAST.

(He looked like a very nice boy.)

As a result, this module outputs the matched list of pronouns given to its input with the antecedents taken form the previous sentence. In the example below; the

possible antecedents in the first sentence that may be matched with the pronouns in the second one are found and the output list is simulated with a figure:

(7.6)

Ali_i ağaçların altında gölde arkadaşlarını$_k$ gördü.

Ali tree.PLU.GEN below.LOC lake.LOC friend.POSS.ACC.3SG. see.PAST.3SG.

(Ali saw [his] firends in the lake below the trees.)

(7.7)

[O_i] Onlara$_k$ el salladı.

They.ACC wave.PAST.3SG.

([He] waved them.)

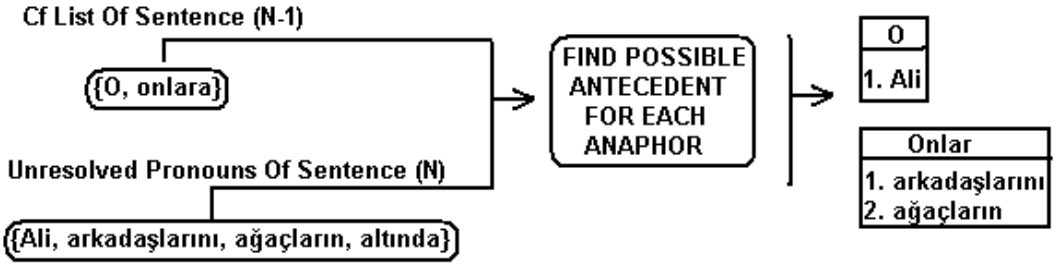


Figure 7.4: Input-Output of Module Find Poss. Ant. Of Each Anaphor

The outputs created by this module will be sent to the nexts one which prepares all of the possible combinations using the alternative choises of the pronouns.

7.1.5 Module: Find Combinations

The sub-module referred to as Find Combinations is responsible for creating combinations resulting from the cross-product of anaphors and their possible antecedents. Let us illustrate this process by an example:

Assume that there are three anaphors in the current utterance such as a_1, a_2 and a_3 and that there are four referring expressions in the C_f list of the previous utterance such as A, B, C and D. Given these inputs, this sub-module generates the pairings shown in Table 7.1:

Table 7.1: An example output of “Find Possible Antecedents” module.

a_1 = {A, B, C}
a_2 = {B, C}
a_3 = {A, B, C, D}

Based on these pairings, we obtain the combinations in Table 7.2:

Table 7.2: All of the possible matching of the unresolved pronouns

	a_1	**a_2**	**a_3**
Combination$_1$	A	B	C
Combination$_2$	B	C	A
Combination$_3$	B	C	D
Combination$_4$	C	B	A
Combination$_5$	C	B	D

Such an output will be sent from this module. Consider the examples given in (7.6) and (7.7) above. If we send the output of “Find Possible Antecedents” for (7.7) input to this module, we will have the following combinations:

Combination$_1$ = {O = Ali, onlara = arkadaşlarını}

Combination$_2$ = {O = Ali, onlara = agaçların }

Then, each of these combinations will be assumed as true and inserted in place of the unresolved pronouns. For every insertion, the transition rule will be recorded. The combination which provides the best transition (cf. Section 5.4.1 for Centering transition hierarchy) will be selected as true by the next module.

7.1.6 Select the Best Combination

A crucial is to decide on which combination provides us with the most likely mapping. This decision is made through the following process. For each combination, a transition case is defined using the Centering rules. The combination which leads to the best transition is selected as the output of this process.

For example, assume that Combination$_4$ above triggers a Continue transition and the others trigger Retain or Shift transitions. In this case, Combination$_4$ will be selected as the best combination since it causes the most salient transition. It means that the antecedent of a_1 will be defined as C, a_2 as B and a_3 as A.

7.1.7 Save Anaphoric Relations

Finally, the Cf list needs to be updated so that it encodes the new mapping values, which are to be used later on. So according the 4 combinations that are used as an

example above, the definiton will be updated and after that time it will be assumed that a_1 refers to C, a_2 refers to B and a_3 refers to A.

7.2 Knowledge Representation

In this application, the language used in the implementation is Java Programming Language. There are many reasons to use this language which are discussed in section (7.4). But the most important one is the object oriented support of this language. This ability provides us to implement our application with seperate objects which makes the whole application as modular.

Since the programming language used in the application is Java, the knowledge is represented by using the classes in the study. Actually the minimal part of the application is thought as the word. So as a result, a word in real sentences is represented as a “Word” type class in our application. In the following, some of the operations and the members of Word type is shown in table 7.3:

Table 7.3: Overall structure of the Word object

Property Name	Type	Description
discourseNo	variable	This variale is used to record the discourse number of the Word object
sentenceNo	Variable	This is used to record the sentence number
orderNo	Variable	This is the order number of the word in the sentence that it belongs to.
caseType	Variable	This variable holds the case information such as; accusative, dative, locative, nominative, genitive, ablative
wordType	Variable	This defines that the word is noun, pronoun, adjective, verb
gramaticRole	Variable	This is the gramatical role of the word. This variable may have the following values: subject, object or adjunct
numberPerson	Variable	This is the number-person information of the word such as; a1sg, a2sg, a3sg, a1pl, a2pl or a3pl
animateType	Variable	This records the animacy information of the word. The animacy type may be human, animal or other. This is useful for semantic approaches.
referTo	Variable	This shows the discourseNo, sentenceNo and orderNo of another word. If it is set, this means that this word is a pronoun and shows another word pointed with this variable
overtness	Variable	This gives the information of whether the word is null or overtly evoked in the sentence
isReferringExpression	Variable	If the word is referring expression, either an antecedent or a pronoun, this variable is set with yes.
setHier	Operation	This is a function that sets the salient weight of the constant variables located in this object. This variables are named as SUB_VALUE, CASE_ACC_VALUE, CASE_LOC_VALUE and etc.. This constant variables of the Word object is used to define the hierarchy.

Of course there are some other methods and variables inside this Word object. In general these are some get and set methods to use the variables shown in table above and some are the internally used inside the object.

By using this basic information, any sentence represented as the “Sentence” class in the application. Actually a sentence may be thought as the array of “Word” objects. So, the Sentence object is already included an array of “Word” objects inside its body. The top level object is the “Discourse” object. This object also includes an array of “Sentence” objects inside and also has its own operations as shown in below. This objects also have some get and set methods which are used for example, to get any word in the sentence, any sentence of any discourse, number of words in the sentence or number of sentences in the discourse and etc.

7.3 The Algorithm By An Example

In this section, the algorithm is presented by an example. Up to this point all linguistic and computational backgrounds are explained in detail.

7.3.1 Necessary Information for The Application

The algorithm that is applied in the application is constructed based on Centering Algorithm mainly, as discussed in previous chapters. And it should be noted that the input and output values shown here are symbolically represented. Since the real representation is technically a Java code, more meaningful representation for the text is choosen to explain the algorithm. According to these temporary representation:

a) An unresolved overt pronoun “A” is represented as:

(A,_,Overt,CASE,PERSON,NUMBER, SENTENCE_POSITION)

b) A null pronoun **A** which is resolved as **AntecedentValue** is represented as:

(A,AntecedentValue,Null, CASE, PERSON, NUMBER, SENTENCE_POSITION)

c) An antecedent **B** is represented as:

(B,B,NonPron,CASE,PERSON,NUMBER,SENTENCE_POSITION)

in the following explanations.

Finally the algorithm stages are presented. These stages are explained by using the following discourse segment:

(7.8)

Ayşe yolda Ali’yi gördü.

Ayşe road.LOC Ali.ACC see.PAST.3SG.

(Ayşe saw Ali on the road.)

(7.9)

[O] Ona elindeki kitapları gösterdi.

S/he.DAT han.POSS.3SG.LOC book.PLU.ACC show.PAST.

([She] showed him the books in her hand)

7.3.2 The Algorithm

Step 1) The algorithm starts with (7.8) as the input.

Step 2) In step 2, the module “Find All Referring Expressions” will provide a list, called C_f(current_sentence) in general, of all antecedents and pronouns, if there exist any. As a result, the following list will be obtained for our example:

CfList(7.8):

(Ayşe, Ayşe, NonPron, Nom, 3rd, singular, subject)

(Ali, Ali , NonPron, Accusative, 3rd, plural, object)

(Yol, Yol , NonPron, Locative, 3rd, plural, object)

Step 3) The unresolved anaphors in C_fList(7.8) will be found by “Find All Anaphors” module in this step. The resulting list will be called urAnaphora(current_sentence). Since, there is no unresolved anaphor in the current utterance, the algorithm will go to step 1 with (7.9) as the input. At the end of the second step, the following Cf list will be obtained:

CfList(7.9):

(O, _, Null, Nom, 3rd, singular, subject)

(Ona, _, Overt, Dative, 3rd, singular, object)

(Kitap, Kitap , NonPron, Accusative, 3rd, plural, object)

The following result will be obtained at the end of step 3 for 7.9 :

urAnaphora(7.9):

a_1: (O, _, Null, Nom, 3rd, singular, subject)

a_2: (Ona, _, Overt, Dative, 3rd, singular, object)

Step 4) For each element a_i of urAnaphora(utterance), the possible antecedents of C_f (previous_sentence) will be filtered according to person and number agreement. As a result, the possible antecedent list will be found, PA(a_i). This process will be performed by filters 4a and 4b in Figure (7.2). The algorithm will produce the following list as a result of step 4:

PA(a_1): {Ayşe, Ali} and PA(a_2): {Ayşe, Ali}

Step 5) For every PA(a_i), the combinations of mapping series of a_i with possible antecedents that are already defined in the previous step will be found. This process will be carried by the module referred to as "Find Combinations". As a result of step 5, the combinations shown in Table (7.4) will be obtained:

Table 7.4: Possible combinations

	a_1	a_2
$Combination_1$	Ayşe	Ali
$Combination_2$	Ali	Ayşe

Step 6) In this step the best combination will be decided upon. This will be done on the basis of generated transitions. For example, a combination leading to a Continue will be preferred to a combination leading to a Retain transition. This step will match the anaphors and the antecedents according to this combination. For our example, at the end of step 6, the best combination will be defined as $combination_1$.

Step 7) Because Cp(7.8), Cb(7.8) and Cb(7.9) do not exist and Cp(7.9) is "Ayşe" (since it is the most ranked element of CfList(7.9)), this mapping will cause a Continue transition according to the "Centering Theory". *$Combination_1$*, however, will cause a "Retain transition". Then, *$combination_1$* will be preferred to $combination_2$ as a result of Rule 1 of the Centering theory.

These relations will be updated in C_f (utterance). So, for our example,

a_1: (O, _, Null, Nom, 3rd, singular, subject)

a_2: (Ona, _, Overt, Dative, 3rd, singular, object)

will be updated to

a_1: (O, Ayşe, Null, Nom, 3rd, singular, subject)

a_2: (Ona, Ali, Overt, Dative, 3rd, singular, object)

Step 8) If the end of the discourse is not reached, the algorithm goes to Step 1 to fetch the next utterance, else it stops

7.4 Technical Tools

In this study, there are a few technical tools that are used to implement the application. Actually one of the main motivation for developing the application is making it flexible that is moved and run in every platform. For this purpose a database is not used for this first approach. Instead, a formatted text file is used as tha database which can easily moved with the program to another machine or platform. The structure of the text file is explained in following.

Before creating the text file, a sheeted file is created in which there is one word and its properties in every line as shown in figure below:

	A	B	C	D	E	F	G	
1	Value	discourse_no	sentence_no	word_order	Animacy	Case	gramatic_role	refe
2	uzaklarda	0	0	0	-1	loc	adj	
3	bir	0	0	1	-1	-1	-1	
4	ada	0	0	2	-1	nom	sub	
5	vardır	0	0	3	-1	-1	-1	
6	orada	0	1	0	-1	loc	adj	
7	hava	0	1	1	-1	nom	sub	
8	yaz_ve_kış	0	1	2	-1	-1	-1	
9	sıcaktır	0	1	3	-1	-1	-1	
10	Ahmet	0	2	0	-1	nom	sub	
11	orayı	0	2	1	-1	acc	adj	
12	çok	0	2	2	-1	-1	-1	
13	sever	0	2	3	-1	-1	-1	

Figure 7.5: An example view of input data sheet

As shown in the figure above, for any word there are number of properties in every line. The whole segments of the propeties can be listed as:

Value: This is the sring value of any word.

Discourse No: Since there are number of discourses in the data file, every word will locate in a strict discourse. This number will give this discourse's id. The first discourse is numbered as 0.

Sentence No: As in discourse no, there will be also a sentence for every number that they are located in. The first sentence of any discourse has the 0 id.

Word Order: The order of the wrods for in a sentence is also numbered as written in the data file. The first word of any sentence has the 0 id.

Animacy: This value is important for semantic decision segments in natural language applications. Even this study is poor in means of semantic modules, this

value is also included in the program for the future purposes. This will give the information of “human”, “animal” or “other” for any word.

Case: This field keeps the case information of the related word. Cases can take one of the “nominative”, “accusative (-i, -ı, -u)”, “dative (-e, -a)”, “locative (-de, -da)”, “ablative” (-den, -dan)” and “genitive (-in, -ın, -un)” values.

Gramatic Role: This is the grammatical information of any word. This filed can take “subj (subject)”, “obj (object)” or “adj (adjunct)” values.

ReferTo: Actually the formatted strings such as “x:y:z” will be located in this field. This means another word whic is located in z_{th} order of y_{th} sentence in discourse z. This data will be directly updated for learning data and will be resolved in the final application for unresolved anaphors.

Number: As its name, this field keeps the number and person information for the related word. The values of this field can be “a1sg (first person singular)”, “a2sg (second person singular)”, “a3sg (third person singular)”, “a1pl (first person plural)”, “a2pl (second person plural)” and “a3pl (third person plural)”.

Overtness: Any word can be overtly written in the sentence or it can be null as discussed in the former sections. This field keeps this overtness information for any word. The field can take “overt” or “null” values.

Referring Expression: Any type of word can have the referentiality property. This words may be pronouns or antecedents. So this information is kept in this field as “yes” or “-1”.

Word Type: Every word, of course, has the type information. A type for any word can be “noun”, “pronoun”, “verb”, “det”.

As seen in figure above, some of the fields are filled with a value –1. This means that for tthe current approach the field value of the related word is ommitted either it is not a suitable word to have the related value or is not used for the current application.

After preparing such a sheet and filling it with the necessary data, the file will be converted a standart .txt file. The final data file is shown in figure below:

Figure 7.6: An example view of final data file.

The most important part of the technical tools is of course the programming language for the application. For this model, java programming language is used because of the reasons that is told below. But the first purpose of choosing the language is the property of its platform independence.

With the help of Object Oriented Structure of Java, the application is easily be handled during the implementation. The resulted sentences are represented graphically and inside a user friendly GUI, for which the reach GUI libraries of Java is used.

Java uses the standart syntax of the languages like C and C++. And also it is one of the leading programming languages that it uses Object Orineted techniques in its nature. Everything is object in Java. All of the codes are starts with a keyword “class” which means that the related part of the code is an object, too. This class that is created by the programmer can be directly used with its suitable properties inside another code segment after they are compiled.

Since the Java Platform is considered the facts and importance of the network communication, the property of platform independence is loaded into the Java environment. This property is provided by a kernel application in Java platform. This application is called the Java Virtual Machine. For any platform, if suitable JVM is installed inside, any Java application can be impemented without changing any single line inside that platform. For example, a java program developed in Unix

platfrom can directly be used and run in Windows environment. As a result, this ability makes the programs easily moved from one machine to another.

When the user starts the application, a user friendly interface is appeared on the screen as shown in figure below:

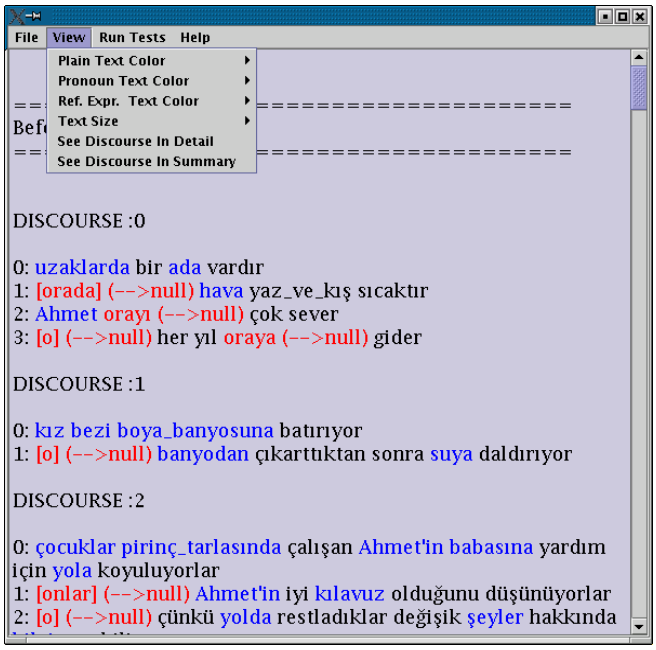
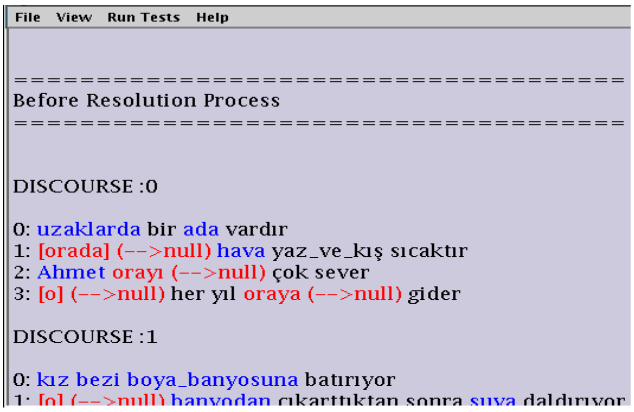


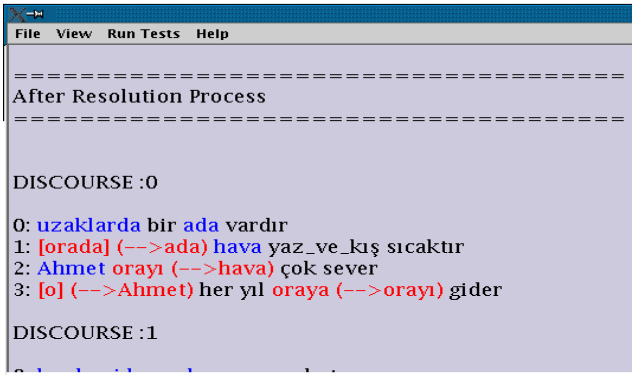
Figure 7.7: The main screen of the application

This interface has some useful menu options to work on the data. As explained above, the data is prepared as the formatted text file in this version. This file is automatically read by the program and according to the choices of the user, some special operations, resolving mainly, is completed and the results are displayed in different ways. In figure 7.8 an example output of the data is shown. In, 7.8 (a) data is displayed before the resolution.

By dragging down the screen the same output can be displayed but this time pronouns and their matching antecedents after the resolution process is evaluated and displayed in the same screen:



a)



b)

Figure 7.8: Discourse information before and after the resolution process

By using the “Run Tests” menu options, it is also possible to run the tests that are used to find the best hierarchy in the applications. And also the algorithm or the implementation can be tested for another criterions without changing the main frame in future. In figure below only the first experiments results ar shown to give a general idea:

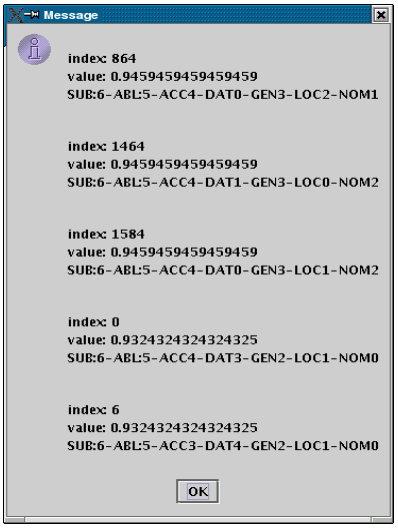


Figure 7.9: Results of the first experiment

Since the application is designed as modular, without changing the main structure it is very easy to add new options, menus or to change the operation of any existing menu option. This provides very extendable and easy-to-use program to the user.

8. The Hierarchy Developed in This Study for Ranking Process

A discourse segment consists of a sequence of sentences U_i, i=1,2,...n. With each sentence U_n is associated a list of forward-looking centers, represented as $C_f(U_n)$, consisting of possible antecedents which are partially ordered according to a number of factors. Ranking of an entity on this list corresponds that it will be the primary focus of subsequent discourse. The first element of the list is defined as preferred center (C_p). The backward looking centers of U_n, denoted as $C_b(U_n)$, represents the entity currently being focus in the discourse after U_n is interpreted. The most important question is how the entities in the list are ordered. A plausible answer could be that the grammatical function plays an important role on this ranking.

Actually the algorithm that should be used is already defined since the basic approach is Centering in this study. The rank hierarchy, however, differs with the properties of the natural language, hence the selection of this hierarchy directly defines the accuracy of the algorithm.

The only study that suggestes a hierarchy for Turkish is Turanlı (1996). This is explained as:

Agent > Experiencer > (Inalienable) Possessor > Theme **(8.1)**

According to the Thematic Hierarchy presented in (8.1), a category of verbs called Psychological verbs, assign an Experiencer theta-role to one of their arguments, either the subject or the object. If the object has an Experiencer theta-role, the subject can receive either an Agent or a Theme role. Experiencer objects rank higher in the C_f list only when the subject has a Theme role. On the other hand; when the subject is an Agent, it ranks higher. It can easily be guessed that, very well implemented thematic application is vital to detect the agent, experiencer and theme roles since it depends on these criterias. Since there was not any presented tools to define such roles for Turkish texts, only the gramatical role and the case marking is assumed in the scope of this study.

On the other hand, other suggested hierarchies in the field are either not for Turkish nor do not based on Centering approach. Hence, the accuracy of the presented approaches are well-tested before suggesting a new one in the scope of the study.

8.1 Experiments on Existing Hierarchies

In the experiments explained below, only the hirerarchies are tested based on the case and grammatical role information because of the reasons told above.

Since the existing hierarchies can not achieve a satisfactory level of results, a new hierarchy is developed and tested on a number of experiments in this study. Consequently, a well hierarchy model is suggested in the scope of this study and it is used in the computational model.

In the first experiment, without taking an hierarchy as the basis, only the natural matching is observed. This first approach is taken and tested on 100 sentences which are taken from Say et al. (2003). For this experiment especially the sentences are selected as the input that include more than one antecedents. The words in these sentences are morphologically analyzed and parsed manually. After that operation, a matching map (78 matchings) according to the case information and the sentence roles, is created for the whole discourse segments in the experiment.

As an example, the matching operation is completed as shown in following sentences:

(8.1)

Kapıdan (abl+sing+adj+overt), içeriye (dat+sing+adj+overt) bir **yabancı$_i$** (nom+sing+sub+overt) girdi.

(A stranger is entered inside from the door.)

(8.2)

Şaşkın bir şekilde **ona$_i$** (dat+sing+obj+overt) bakıyorduk.

(We were looking at him suprisingly.)

Result: “dat+sing+obj+overt” matches “nom+sing+sub+overt”

This matching process is also completed manually and as a result, the following table has found at the end of the experiment:

Anaphors					Antecedents			Number of occurrence		Percentage
	Role	Case	Number		Role	Case	Number			
overt	adj	abl	sing	matches	obj	acc	sing	1		1,28
overt	adj	dat	plu	matches	sub	nom	plu	2		2,56
overt	adj	dat	sing	matches	sub	nom	sing	1		1,28
overt	obj	acc	sing	matches	sub	nom	sing	1		1,28
overt	obj	acc	sing	matches	obj	abl	sing	1		1,28
overt	obj	acc	sing	matches	obj	acc	sing	4		5,12
overt	obj	dat	sing	matches	sub	nom	sing	1		1,28
overt	obj	dat	sing	matches	obj	acc	sing	2		2,56
overt	obj	lok	sing	matches	obj	acc	sing	1		1,28
overt	poss	gen	sing	matches	obj	acc	sing	2		2,56
overt	sub	acc	sing	matches	sent	nom	sing	2		2,56
overt	sub	gen	sing	matches	obj	acc	sing	1		1,28
overt	sub	nom	sing	matches	obj	acc	sing	3		3,84
overt	sub	nom	sing	matches	sent	nom	sing	1		1,28
null	sent	nom	sing	matches	sub	nom	sing	1		1,28
null	sub	acc	sing	matches	sub	nom	sing	1		1,28
null	sub	nom	plu	matches	sub	nom	plu	3		3,84
null	sub	nom	plu	matches	sub	acc	plu	1		1,28
null	sub	nom	plu	matches	sub	nom	plu	1		1,28
null	sub	nom	plu	matches	adj	dat	sing	2		2,56
null	sub	nom	plu	matches	sub	nom	sing	2		2,56
null	sub	nom	sing	matches	adj	dat	sing	2		2,56
null	sub	nom	sing	matches	sent	nom	sing	5		6,4
null	sub	nom	sing	matches	obj	acc	sing	6		7,68
null	sub	nom	sing	matches	sub	nom	sing	31		39,68
							Total	78		

Figure 8.1: The resulting matchings of the first experiment

As it can easily be observed that there is not any strict result about the matching which directly bases on the sentence role and case information. For example; we cannot say anything for any word whose property information is overt+obj+acc+sing, since it matches with sub+nom+sing for 1 time and with obj+abl+sing for 1 time and obj+acc+sing for 4 time as shown in Figure (8.1). The only highlited matching is sub+nom+sing with sub+nom+sing located in the last line of the figure. This result, however, is not enough to create a whole hierarchy for all roles and cases.

In the second experiment, the hierarchy presented in Kılıçaslan (2004) is tested for the suitability.

Actually this hierarchy is not based on the Centering Theory, but it is for defining the specifity criteria which is told in the same study in detail.

According to this suggestion, the hierarchy is given as:

Inc. > Non-Inc. and Non-Case-Marked > Acc. Marked > Non-Acc. Marked **(8.2)**

By assuming the Situtation Theoretic Approach given in Kılıçaslan (2004), which defines the following hierarchy:

loc < abl < nom < dat < acc **(8.3)**

the final suggestion can be taken as:

loc < abl < nom < dat < acc < unmarked < incorporated **(8.4)**

In this study, 100 sentences which especially have more than one antecedent inside are taken from Kurtuluş Yayınları (1990). This time, the matching frequency of the most suitable (the most salient) antecedent of the first sentence with a pronoun in the second one is provided. By the help of the results, a salient matching procedure is seeked.

In the example below, this manual matching operation is illustrated:

(8.3)

Ali Baba'nın sesi mağaranın duvarında yankılandı.

Ali Baba'nın - gen

sesi – acc

mağaranın – gen

duvarında - loc

(8.4)

Bu ses ona cesaret verdi.

Result: ona matches Ali Baba; **gen > acc**, **gen > loc**

The results of this experiment taken from the data file is given in figure 8.2:

Belong to modifier				Belong to modifier	Time
	Acc	**>**	**Abl**	**True**	**1**
	Acc	>	Acc	True	3
True	Acc	>	Acc	True	1
	Acc	>	Acc		2
	Acc	**>**	**Dat**	**True**	**2**
True	**Acc**	**>**	**Dat**	**True**	**1**
True	**Acc**	**>**	**Dat**		**1**
	Acc	**>**	**Dat**		**1**
	Acc	>	Gen	True	1
True	Acc	>	Gen	True	1
	Acc	>	Gen		4
True	**Acc**	**>**	**Nom**	**True**	**1**
	Acc	**>**	**Nom**		**7**
True	**Acc**	**>**	**Nom**		**1**
	Dat	>	Acc		1
	Dat	>	Gen	True	1
	Dat	**>**	**Nom**		**7**
	Dat	**>**	**Abl**		**1**
	Dat	>	Acc	True	2
	Gen	>	Abl	True	1
	Gen	>	Acc		3
	Gen	>	Dat	True	1
	Gen	>	Dat		1
	Gen	>	Gen	True	2
	Gen	>	Loc	True	1
	Gen	>	Nom		7
	Nom	>	Acc	True	2
	Nom	>	Acc		5
	Nom	>	Dat		3
	Nom	>	Gen	True	1
	Nom	>	Gen		3
	Nom	**>**	**Loc**	**True**	**2**
	Nom	**>**	**Loc**		**1**
	Nom	>	Nom		1
				Total:	73

Figure 8.2: Results of second experiment

In these results, the bold lines means that the selected hierarchy is provided. So if we look at the results it can easily be seen that the there are very few true result is occured in this experiment. As a result, the second approach is also eleminated for our application since the number of provided cases is not salient among the whole selected discourse.

8.2 Experiments on New Suggested Hierarchies

In the next two experiments, inspiring the former hierarchies, a new hierarchy model is suggested which bases only on the subject role and case marks. According to this

new suggestion the salience criteria is based on criterias = sub, acc, dat, loc, abl, gen and nom. For the purpose of finding the dominance criteria among them, three new algorithms are implemented and tested on the test data.

In the first algorithm (named as Test B on program's menu) all of the pronouns of the sentences in iearning data is observed. In this observation all of the antecedent informations which are referenced by a subject pronoun are recorded such as; if any antecedent has the criteria x that is one of the properties from the criterias above, the value of the criteria is increased with 1.

The basic assumption in the experiment that if there is a null subject pronoun in a sentence, it will show the most ranked element in the sentence before it (Turanlı, 1996). By using this assumption, Only the sentences are taken into the evaluation which has null pronoun in subject position.

For example, since the selected antecedent role is subject in the first sentence below, a variable that records the number of the subject is increased while the other ciriteria variables remain the same:

(8.5)

Ali_i okula gidecek.

Ali school.DAT. go.FUT.3SG.

Ali will go to the school.

(8.6)

[O_i] Orada derse girecek.

There.LOC. lecture.DAT. attend.FUT.3SG.

He will attend the lucture there.

All of the values starts from 0 before the test. And after the test is applied, the results are observed.

This test is applied on totally 23 discourses each has averagely 4 or 5 sentences. There are totally 330 referring expressions in these sentences and 115 of them are pronouns.

At the end of the test, the results shown in the table are observed. According to these results, the most salient element is generally "subject" in Turkish texts.

Table 8.1: Results of the first test

Hierarchy Criterion	Number of Points by Null Subject Pronouns
Subject	63
Dative Case	3
Genitive Case	2
Accusative Case	1
Ablative Case	0
Locative Case	0

In the second algorithm (Test A on the program's menu) the approach is a bit different from the first time. Before starting the test, a code is developed to create all of the different ordered combinations of the criterias. For example, only the 4 combinations that this code can result is given in below:

(8.7)

sub > acc > loc > abl > nom > dat > gen

sub > loc > acc > abl > nom > dat > gen

acc > loc > sub > abl > nom > dat > gen

gen > acc > sub > abl > nom > dat > loc

...

Using every single result of these combination and assuming that it is the ranking hierarchy, all of the learning data is evaluated and for every combination, the accuracy of the results are recorded. Finally the best 5 combination which causes to have the best accurate result is found as the following:

Table 8.2: The best five accurate hierarchies found by the second test

Hierarchy	Accuracy
Sub > Abl > Acc > Gen > Loc > Nom > Dat	0.94
Sub > Abl > Acc > Gen > Nom > Dat > Loc	0.94
Sub > Abl > Acc > Gen > Nom > Loc > Dat	0.94
Sub > Abl > Acc > Dat > Gen > Loc > Nom	0.93
Sub > Abl > Dat > Acc > Gen > Loc > Nom	0.93

This second algorithm is also tested on the same learning data.

Note that the results may change if the amount of the learning data is increased, of course. But the current one is also sufficient since two experiments seems to have similar results.

As a consequence, assuming the test results and comparing them with the old tests, the following hierarchy is selected and suggested for Anaphora resolution process based on the Centering Theory on Turkish texts:

SUB > ABL > ACC > GEN > LOC > NOM (non-subject) > DAT **(8.5)**

As it is explained in section 7.1.2 this hierarchy is direcly used in Module Ranker in our computational approach.

8.3 The Results

The test data and the learning data consist of same type of sentences in general. The level of the sentences are on primary school level and they are taken as having more than one antecedent inside. By this way, more chance to make enough comparison between the antecedents has been provided.

And also the sentences are carefully selected to be more than one antecedents in source sentences.

As the final observation anaphora Resolvent for Turkish has worked with approximately %94 of accuracy among these test data.

According to the results produced by the tests, it is observed that the subject role is generally more salient in Turkish sentences similarly in English. On the other hand, the hierarchy among the remaining cases may differ depending on the learning data. However, according to the results of tests explained in section 8.1, the salience is not dependent on the expressions's cases according to each other which are located in the same sentence. But it can be defined by a general hierarchy as done in this study. It is observed that the hierarchy has the similarity with the one used in this study. And also it is observed that the null subject pronoun is generally points an antecedent which is in the subject position.

9. Conclusion and Discussion

In this study a new model based on the Centering Theory (Grosz et. al 1995) is developed that resolves the pronominal anaphors in Turkish. All of the necessary modules required for such a model is designed in the scope of the study. And many parts are also implemented to observe the results. Since the basic approach is the Centering, it has to be developed a ranking hierarchy for implementing the application. For this purpose, the existing hierarchies suggested in former studies are tested and observed whether they are suitable for using in this study. It is observed that very few of them are specific for Turkish. The hierarchy suggested in Turanlı (1996) involves the thematic roles. For this reason, some well designed tools are needed to define these roles in the sentence. And similarly, the hierarchy suggested in Kılıçaslan (1994) also includes thematic roles. Unfortunately there was no tools for defining such roles in the time of developing this study. For these reasons, it is not possible to assume such type hierarchies includes the thematic roles to implement the application.

The most basic hierarchy for English is presented in Brennan (1987). Since only the gramatic roles are the criteria in that hierarchy, with all of the other gramatic roles (subject, object, adjunct) the suitability of this hierarchy is tested on a Turkish discourse. In this test 100 sentences are taken and a gramatical matching rules are observed between the pronouns and their antecedents. But the observation is that the gramatic roles can not define the salience factor in Turkish directly. The second observation is tested on the hirerarchy suggested in Kılıçaslan (2004). This hierarchy is based on the case-marking and actually suggested to define the specifity in Turkish. Anyway, it is tested whether it can be used also for ranking process in Centering algorithm which tries to define the most salient element among all of the referential expressions. Hence, new tested are completed on 100 other sentences to observe the accuracy of this approach. At this time sentences are selected including more than one referential expression inside and one or two pronouns in the sentences following them. And the most salient is defined and its case is recorded as more salient among the others which are in the same sentence.

As a result, it is observed that the hierarchy depending only on the case marking is not suitable for Turkish, either.

So, after the observations above, it is decided to suggest a new hierarchy in which case marking and grammatical roles are included. Using the following hierarchy founded in the study, approximately %90 of accuracy is observed on a test data including 100 sentences:

SUB > ABL > ACC > GEN > LOC > NOM (non-subject) > DAT

Hence, needed hierarchy defines the most salient referential expression based on the case marking and the gramatic roles (subject only) is suggested and used in the scope of this study.

For a system which resolves the anaphors, many other modules besides the one makes the ranking, are needed to be developed. Two of them are the tools completes the operation morphologic analyzing and the parsing. Since the development of these modules is not in the scope of the study, the existence of such tools are seeked. Morphologic Analyzer (Oflazer ,1994) is used for the morphological analyzing process. The parsing operation, however, is completed manually since there was not any suitable tool for Turkish at the time of writing the study. The entegration of such a tools to the application can be seen as a future work at this level.

Additionally, defining the referential expressions and the pronouns and the relation between them is also very important task to develop an anaphora resolution system. This definition, of course, needs some linguistic findings and rules used for modeling the operation. Many of the needed findngs for Turkish are taken from Turanlı (1996). The rules that are suggested theorically in that study, is used in the other modules of this study and an extensive computational model is designed in this study.

Of course, there is much to do to develop a computational system that carries out anaphora resolution to a desirable extent. Even though our approach cannot be counted as simplistic, it still needs to be improved in various dimensions. A dimension along which this work can be further developed relates to pure anaphors. This, of course, requires a satisfactory linguistic account of the behaviors of these anaphors in Turkish.

Genericity is another issue awaiting for an explanation in terms of how it interacts with anaphora resolution. As should have been noted, this level of interpretation is ignored in our study. Also, it is needless to say that more semantic, pragmatic or

syntactic constraints need to be incorporated to the system in order to generate more accurate results.

REFERENCES

Kuno, S., 1970. Some Properties of Nonreferential Noun Phrases, in R. Jacobson and S. Kawamoto eds., Studies in General and Oriental Linguistics.

Bar-Hillel, Y., 1971. Language and Information. Selected Essays on Their Theory and Application, Addison-Wesley, Reading, Mass.

Givón, T., 1976. Topic, pronoun, and grammatical agreement. In C. Li (ed.), Subject and topic, New York Academic Press, 149-188.

Halliday, M. and Hasan, R., 1976. Cohesion in English, London: Longman

Grosz, B., 1977. The Representation and Use of Focus in Dialogue Understanding Technical Report, 151, SRI International, 333 Ravenswood Ave., Merlo Park, CA., 94025, USA.

Sidner, C., 1979. Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse, Technical Report, 537, Artificial Intelligence Laboratory, Canbridge, Massachusetts Institute of Technology.

Webber, B., 1979. A Formal Approach to Discourse Anaphora, Garland, New York.

Chomsky, N., 1981. Lectures on Government and binding. Foris, Dodrecht.

Prince, E. F., 1981. Toward a Taxonomy of Given / New Information in P. Cole ed. Radical Pragmatics, Academic Press, New York.

Fromkin V. and Rodman R., 1983. An Introduction to Language. Holt, Rinehart & Winston, New York.

Givon, T., 1983. "Topic Continuity in Spoken English" in T. Givn (ed.) Topic Continuity in Discourse: A Quantitative Cross-Language Study. Philadelphia: John Benjamins Publishing Company.

Grosz, B., Aravind J. and Scott W., 1983. Providing a Unified Account of Definite Noun Phrases in Discourse, In Proc 21st Annual Meeting of ACL, Assoc. of Computational Linguistics.

Erguvanlı, Eser, 1984. The function of Word Order in Turkish, University of California Press, Los Angeles.

Grishman, R., 1986. Computational Linguistics An Introduction, Cambridge University Press, Melbourne, Australia.

Dede, M., 1986. Definiteness and Referentiality in Turkish Verbal Sentences, in D. Slobbin and K. Zimmer eds, Studies in Turkish Linguistics, John Benjamins, Amsterdam.

Brennan, S., Marilyn W. F. and Carl P., 1987. "A Centering Approach to Pronouns" in Proceedings of the 25th Annual Meeting Of ACL, Stanford. Fox, B., 1987. Discourse Structure and Anaphora. Cambridge: Cambridge University Press.

Mitkov, R., 1988. “Robust Pronoun Resolution With Limited Knowledge”, *Proceedings of the 17th International Conference of Computational*

Linguistics (COLING'98/ACL'98), 869-875. Montreal, Canada.

Hankamer, J., 1989. Lexical Representation and Process. Chapter Morphological Parsing and the Lexicon. The MIT Press.

Ingria, R. and Stallard, D., 1989. A computational mechanism for pronominal reference. *Proceedings of the 27th Annual Meeting of the ACL, 262-271*. Voncouver, British Columbia.

Kuno, S., 1989. Identification of Zero Pronominal References in Japanese Unpublished ms.

Kurtuluş Yayınları, 1990. Ali Baba ve Kırk Haramiler, Kurtuluş Ofset Basımevi, 1990- Ankara.

McCord, M., 1990. Slot Grammar: A System for Simpler Construction of Practical Language Grammars. *In Natural Language and Logic: International Scientific Symposium. (Ed: Studer, R.), pp. 118-145.* Lecture notes in Computer Science, Berlin: Springer-Verlg.

Enç, M., 1991. The Semantics of Specificity, Linguistic Inquiry, 22.

Ofllazer, K. and Kuruöz, İ., 1994. Tagging and morphological disambiguation of Turkish Text. *In Proceeding of the 4th Applied Naitonal Language Processing Conference*. Pages 144 149. ACL, October

Bird, S., 1994. Computational Linguistics, 20 (3), University of Edinburgh.

Haegamen, L., 1994. Introduction to government and binding theory, Oxford: Blackwell.

Lappin, S. and Leass, H., 1994. An Algorithm for Pronominal Anaphora Resolution, *Computational Linguistics*, **20 (4)**, 535-561.

Kılıçaslan, Y. , 1994. Information packaging in Turkish. *Unpublished MSc. thesis*, University of Edinburgh, Edinburgh.

Oflazer K., 1994. Two-level Description of Turkish Morphology, *Literaryand Linguistic Computing*, **Vol. 9**, No:2.

Chomsky, N., 1995. The minimalist program, MIT Press.

Grosz, B., Aravind J. and Scott W., 1995. "Centering: A Framework for Modeling the Local Coherence of Discourse", Computational Linguistics.

Carvalho, A., 1996. "Logic Grammars and Pronominal Anaphora". Proceedings of the 'Discourse Anaphora and Anaphor Resolution' Conference (DAARC'96), 106-122. Lancaster,UK.

Kennedy, C. and Boguraev B., 1996. "Anaphora for Everyone: pronominal anaphora resolution without a parser", *Proceedings of the 16th International Conference on Computational Linguistics* COLLING'96), 113-118. Copenhagen, Denmark.

Turanlı, Ü. D., 1996. Null Vs. Overt Subjects in Turkish Discourse: A Centering Analysis, *Ph.D. Dissertation*, University of Pennysilvania, Philadelphia.

Yüksel, Ö., 1997. Contextually Appropriate Anaphor/Pronoun Generation For Turkish,. *Master Thesis*, Middle East Technical University, Ankara.

Manning D.C. and Schütze H., 1999. Foundations Of Statistical Natural Language Processing, Massachuetts Institute of Technology, USA.

Geman, S. and Johnson, M., 2000. Probability and Statistics in Computational Linguistics, a Brief Review. DRAFT of December 7. Brown University.

Jurafsky, D. and Martin, J.H., 2000. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, Inc., New Jersey.

Hausser, R., 2001. Foundations of Computational Linguistics, Human Computer Communication in Natural Language, Springer-Verlag Berlin HeidelBerg.

Oflazer, K., Hakkani-Tür, D. Z. and Tür, G., 2002. Statistical Morphological Disambiguation for Agglutinative Languages, Computers and the Humanities, Vol. 9 No: 2.

Say B., Oflazer K., Atalay N. B., 2003, Building and Exploiting Syntactically Annotated Corpora (Building a Turkish Tree Bank), Kluwer Academic Publishers, London.

Kılıçaslan Y., 2004. A Situation Theoretic Approach to Case Marking Semantics in Turkish, Trakya University, Edirne, Turkey.

Appendix A. Source Code Of The Application

It can be found on the CDROM.

Appendix B. Dictionary Of Terms

A

Ablative : İsmin -den hali
Accusative : İsmin -i hali
Agent : Olaya sebep olan unsur
Ambiguity : Anlam belirsizliği
Anaphora : Anafora, gerideki kaynağı taşıyan öğe
Antecedent : Önce gelen, kaynak, gösterici kaynağı
Artificial : Yapay

C

Case : Hal durum eki, hal
Cognitive : Bilmeye, kavramaya ya da idirak etmeye ilişkin şey
Coherence : Tutarlılık
Cohesion : Uyum
Context : Bağlam

D

Dative :İsmin –e hali
Definite : Belirgin, açık
Discourse :Metin, Paragraf

E

Entity :Varlık
Expression : İfade

G

Gender : Gramerdeki cins bilgisi
Generic : Genel, Tür
Grammar : Gramer, dilin temel kuralları

K

Knowledge : Bilgi

L

Lexical : Sözlüğe ait, sözcüklere ait.

Lexicon : Sözlük

Locative : İsmin –de hali

N

Nominative : Ismin yalın hali

P

Phenomenon : Olay, Hadise, Olgu

Pragmatics : Anlamak ya da idirak etmenin pratiğe (gramer) bağlı durumu

Predicative : Doğrulayıcı, yüklemi oluşturan

Process : Süreç, işlem

Pronoun : Adıl

Pronominal : Adıla ait

R

Reference : Gösterici, atıf

Referent : Gösterilen, atıfta bulunulan

Referring expression: Gösterici ifadesi, atıfta bulunan ya da bulunulan ifade

Reflexive pronoun : Dönüşlü ya da eylem gösteren adıl (kendim, kendisi gibi)

Resolve : Çözmek, çözümlemek

S

Salience Weight :Baskın olma ağırlık katsayısı

Semantics : Anlambilim

T

Theme : Tema, konu

Token : Bir parçanın en küçük alt birimi (Kelimeler)

U

Utterance :Genel yapıyı oluşturan parçalar (Cümleler)

V

Vocabulary : Kelime hazinesi, kelime dağarcığı

PERSONAL HISTORY

R. Erman Aykaç was born in 26/02/1979 in Mersin. He started his university education in 1997 at Istanbul University Electronics Engineering Department. After his graduation, he started to work as a research assistant in Bilgi University, Deparment of Computer Science in August 2001. In 2002, he started his Msc. Program at Istanbul Technical University in the Deparment of DefenceTechnologies. Erman Aykaç has been working in Bilgi since August 2001.